80-U.S.

$3.00 PER COPY

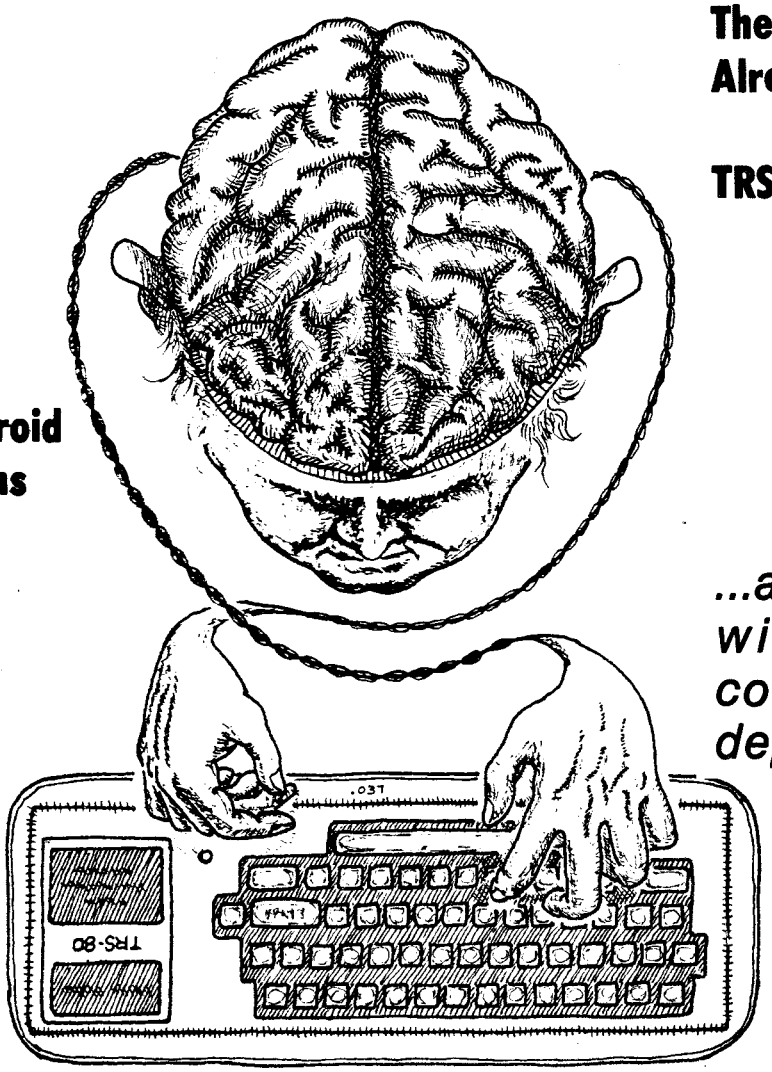
The TRS-80 Users Journal

Volume II, Number 3 May/June 1979

WHICH BRAIN?

The Monitor You Already Have

TRS80 FORTRAN
an evaluation



Chinese, Android and other Nims

"Gee Whiz"

Mind Reader

...along with regular columns and departments

JADE HAS WHAT YOUR TRS-80 NEEDS!

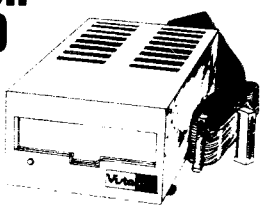
Announcing the easy way to add 16K!

Jade's memory expansion kit contains everything you need to **add 16K** of memory to **either your keyboard or expansion interface** in just 15 minutes. Kit includes memory chips, DIP programming blocks, protective anti-static material, and concise, easy to follow instructions. Jade unconditionally backs this kit with a 90 day warranty on all parts.

Price
$75.00

Vista V80 Mini Disk System for TRS-80

Introducing the
Vista V80 Mini Disk System



23% MORE STORAGE CAPACITY–
Increases your usable storage capacity 23% from 55,000 to 67,800 bytes on drive one.

FASTER DRIVE–
Electronically equal to the TRS-80 Mini-Disk System, but up to 8 times faster (track-to-track access in 5ms for the V80 versus 40ms for TRS-80).

HERE'S WHAT YOU GET:
Minifloppy disk drive/Power Supply/ Regular board/Compact case/Cable.

DOUBLE DENSITY FOR DOUBLE STORAGE–
The V80 will work with the Vista double-density expansion unit when available.

SHIPPED TO YOU READY TO RUN–
Simply take it out of the box, plug it in and you're ready to run.

Special Introductory Price:

$395.00
INCLUDES CABLE

Requires 16K Ram, Level II Basic, Ex. R.

Software for the TRS-80

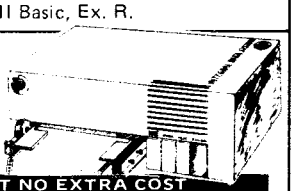
THE LIBRARY 100 FROM TBS, INC.
100 very useful Level II Programs on 5 Cassettes, packaged in an attractive case, all for only **$49.95**

THE ELECTRIC PENCIL FOR TRS-80 FROM SHRAYER SOFTWARE
The time-tested word processing software package package now adapted for TRS-80 owners (Level II). **$99.95**

MICROCHESS FROM SMALL SYSTEM SOFTWARE
Make your TRS-80 (Level II) into a pretty good chess player
Only **$19.95**
Other programs available include: **Game of Life $14.95**
Air Raid $14.95
ESP-1 (8080 Assembler/Editor) **$29.95**

AVAILABLE SOON: Many more programs, plus a complete Software Catalog.

Integral Data Systems



LOADED WITH EXTRAS AT NO EXTRA COST

• Microprocessor controller • Serial RS232C Interface • Parallel TTL level interface • Full upper and lower case ASCII character set (96 characters) • Full 8½" inch wide paper • Line length of 80 columns at 10 characters per inch • Impact printing • 7x7 dot matrix • Ordinary paper - roll, fanfold, or sheet • Serial baud rate to 1200 bits per second • Multiple line buffer of 256 characters • Instantaneous print rate to 100 characters per second • Multiple copies without adjustment • Reinking ribbon mechanism • Front panel operator controls • Attractive table top console

IP-125 Integral Data System IP-125 Friction Feed Printer
- 96 upper & lower case ASCII character set
- Enhanced character control
- Serial RS232C Interface, std. wiring, Parallel TTL optional
- 80 column line, 256 byte multiline buffer

$799

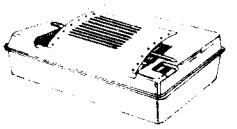
IP-225 Integral Data System IP-225 Tractor Feed Printer
- All standard features of IP-125
- Tractor feed paper drive
- Forms Control Option (P1250)

$949

TRS-80 Interface and Cable $49.00

EXPANDOR'S BLACK BOX PRINTER

This 64-character ASCII impact printer with 80-column capability is portable and uses standard 8½" paper and regular typewriter ribbon. Base, cover and parallel interface are included. Assembled and complete with manual and documentation. **only $470.00**



(90 day manufacturer's warranty)

TRS-80 Interface Cable for Black Box Printer with mating connectors: **$48.00**
(must be used with expansion module, +8v 1 amp power supply required.)

Power Supply for TRS–80/Black Box Printer **$49.00**

JADE
Computer Products

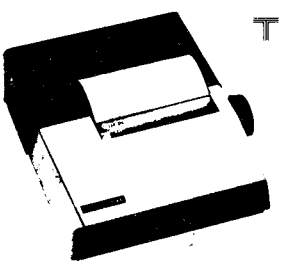
4901 W. ROSECRANS AVENUE
Department "M"
HAWTHORNE, CALIF. 90250
U.S.A.

Telephone:
(213) 679-3313
(800) 421-5809 Continental U.S.
(800) 262-1710 Inside California

Cash, checks, money orders, and credit cards accepted. Add freight charge of $2.50 for orders under 10 lbs. and $1.00 service charge for orders under $10.00. Add 6% sales tax on all parts delivered in California. Discounts available at OEM quantities.

WRITE FOR OUR FREE CATALOG
All prices subject to change without notice.

TRENDCOM 100
Intelligent Printer



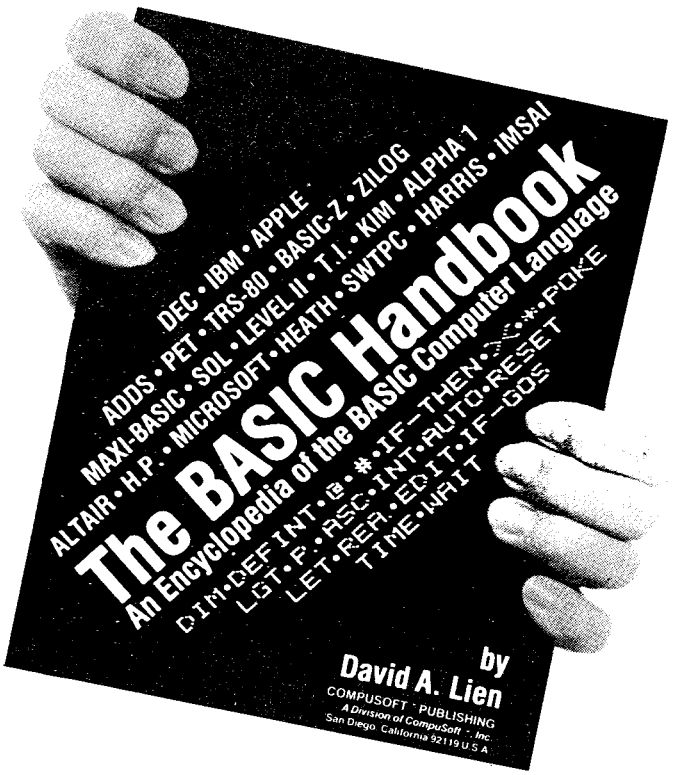
$345

- 40 character per second rate
- Low cost thermal paper
- 96 character set
- Microprocessor controlled
- Bidirectional look-ahead printing
- Quiet operation
- No external power supplies
- Only two driven parts
- High reliability
- Clear 5 x 7 characters
- Attractive metal and plastic case

TRS-80 Interface Cable $45.00

Apple II Interface & Cable $60.00

CompuSoft™ Publishing Proudly Announces . . . . . .



$14.95

(360 pages - Soft Cover)

From the author of the highly acclaimed TRS-80 Learner's Manual comes the book you've been asking for! THE BASIC HANDBOOK is THE definitive reference and "idea" book which explains in detail the BASIC language as used in over 50 micros, minis and mainframes.

Not a dictionary, not a text, it is a virtual ENCYCLOPEDIA of the BASIC language. Explaining all you need to know about over 250 BASIC statements, functions, operators and commands, it is the "missing link" needed to convert programs from one computer to RUN on another.

YES, I need the BASIC Handbook!

COMPUSOFT PUBLISHING 8662 Dent Drive, San Diego, CA 92119

Please send _____ copies of The BASIC Handbook. My check for $14.95 each + $1.35 P&H is enclosed. (CA Addresses add 6% tax)

I understand my order will be shipped promptly and there is a 30 Day money-back guarantee.

Name _____

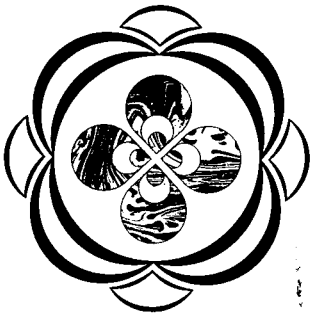
Address _____

City _____ Zip _____

My computer is a _____

CL 80

RANDOM ACCESS



BACKSPACES

We're going to start off the column this issue with a couple of backspaces - corrections, if you will, and an amplification or two. In the Jan-Feb 79 issue, our article entitled "RENEW that PROGRAM" assumed that the number of files question was answered by default. The beginning of the basic program is at 68BA and 68BB only when the number of files is 3, i.e., the default value. Otherwise, it will move, depending on how many files you specify.

Also in that issue there are two corrections to be made to the program listing with the article MATH DRILL ala KING Line 460 should read: 460 IF C=A*BT.1000 and line 2550 should read: 2550 N.N(not N.M). This problem with the Level I listings should be gone from now on. We have now the capability to read Level I onto disk, run it and print out to the line printer.

In the Mar-Apr 79 isse we mentioned in the LETTERS that the word "THEN" can always be replaced by a comma. It certainly can, Ollie - but, there is a "GOTCHA" involved. Seems that the Radio Shack renumber program will not see that comma as a "THEN", and so it leaves the line number following it unchanged.

Again, in the Mar-Apr issue, Figure 1 on page 44 is missing the letter "b" and its corresponding number, 130. It was in the original manuscript George sent us, but must have ended up on the cutting room floor or something.

SURVEY TAKEN

The Orange County TRS-80 User Group newsletter, OCTUG, for March 79 printed the results of a survey they took, which is very interesting. Seventy members responded to the survey, and some of the information gleaned from this survey follows: 83% never owned a personal computer before owning a TRS-80; 68% have Level II; 45% do not have a printer; a whopping 52% have disk systems; 60% said they use their computers for business; 58% for word processing; 89% said they were generally happy with their choice of computer, while 9% said they were not. Very interesting!

We just purchased an Expansion Interface for our second system (backup). It has serial number 26,XXX, while our first one, over a year old, has number 474. We have been told that someone in our area just received a new keyboard with a serial number above 180,000! And in a March 79 Press Release, Radio Shack announced Tandy Corp sales of $66,565,000.00 for February 1979, a 7% gain over February last year. (Add your own ending to this item!)

WHAT IS AN ANDROID?

The question has been raised, since ANDROID NIM is now enjoying a phenomenal popularity, as to just what is an Android? How does it differ from a Robot? or a Cyborg? Obviously, Android is a Science Fiction term, so you won't find it in your Funk & Waggnal. Maybe Asimov or Heinlein would know - any comment from our Sci-Fi readers?

IT'S GOT TO BE FLATTERY

If imitation is the sincerest form of flattery, we should be flattered. Kilobaud magazine for March 79 has an article on the "Keyboard Interrupt for the TRS-80", something we explored in detail in our very first issue in Sep-Oct 78. The same publication in their March 79 issue has an article entitled "Free Speech Lessons for the TRS-80", covering the identical subject we carried in Jan-Feb and called "An Audio-Visual Demo". Which just goes to show that when we say "You saw it first in the JOURNAL", we mean it!

UPPER/lower CASE

Did you know that with the Selectra-Print you can have upper/lower case without a hardware mod? Simply type your lines to be output to the printer using normal typewriter action, i.e., shift for upper case. It will not show on the screen (it will show all upper case). But when you go to print out, throw the little toggle switch on the interface box (which comes with the Selectra-Print) to the TRS-80 position, and you will have normal upper/lower case!

NEW PRODUCTS

The most sensational item around here lately has been a thing called "NEWDOS+", from APPARAT, Inc. 6000 E. Evans Ave., Denver, Co. 80222. It is a TRS-80 DOS, with most of the previous problems fixed, and a host of new things added to make it superb! Just for openers, would you like to do a DIR while in Basic and then return to Basic and find it still there? How about a renumber which you can call from disk basic? Or how about asking for a Reference list of all the variables in your basic program, listed on the screen or on the printer? How about auto key de-bounce and printing the screen to the printer just by holding down the JKL keys? These and many more features are in the NEWDOS portion. The "PLUS" portion adds the following utilities:

DIRCHECK - Test and list Disk Directory
DISASSEM - Machine code Z80 disassembler,
EDTASM - with improvements, Tape and Disk I/O
LEVEL I in Level II - save Level I programs on Disk
LMOFFSET - Machine code Tape/Disk Transfers

Last but not least, (this is the real jewel) SUPERZAP - this utility (written in basic) allows you to instantly look at any memory block or Disk Sector. That's not all - it lets you modify what you see and write it back out - even to disk! With this you can take apart just about anything, and we have already "repaired" a bad directory track. All we can say about it now, having had it for only a short time, is that it's great. We will cover it in more detail in the upcoming issues. NEWDOS and NEWDOS+ are available from Miller Microcomputer Services, 61 Lake Shore Road, Natick, Ma. 01760. NEWDOS costs $49.95 and NEWDOS+ is $99.00; add $1 to either for shipping and handling and Ma. residents add the Governor's 5%.

NEW LINE OF SOFTWARE

Mervin G. Faulkner Assoc., Ltd., 11 Mill St., Hillsdale, Ontario, Canada LOL 1VO, have announced a new line of Software for TRS-80 Level II and 32K Disk Systems which include a Financial Budget with Inquiry for L2 16K at $19.95 and General Ledger, Accounts Payable (with check writing), Accounts Receivable (with invoice writing), Payroll and Inventory, all for 32K disk systems; all include their Inquiry feature. They also have specialty packages for Dentists, Doctors, Real

Estate, etc. Write to Ross W. Gough, Mgr. Micro-Division.

NEWS RELEASE

Electronic Specialists inc., 171 South Main St. Natick, Ma. 01760 have released information on their ISOLATOR, which has three individually filtered 3-prong AC sockets with Integral Surge Suppression. It connects to a 120 VAC line with a standard 3-prong plug, and can accomodate an 1875 watt load. It is called the MODEL ISO-1A and costs $49.95.

ELECTRIC SECRETARY

The Electric Secretary is a New and Noble concept in word processing, announced by The Peripheral People, PO Box 524, Mercer island, Wa 98040. It requires a minimum of 32K and has rapid file access and a unique hyphenating dictionary. The system "builds" a library of hyphenated words, learning as it grows, and has file coupling, which permits lengthly manuscripts to be prepared without overloading memory. The TRS-80 Electric Secretary is supplied on formatted Disk and priced at $75.00 postpaid.

NEW PROGRAM

Computer Generated Data, 638 Muskogee Ave., Norfolk, Va. 23509 offers a Disk Based Mailing/File Label Print program called ULTIMAIL ™. It operates under all versions of TRSDOS and packs 1000 files on a system disk in a one-drive system. It includes HAL™, a program monitor system which protects all data against loss. It has user-controlled spacing between labels and print quantity so that the number of each label may be specified. It is available from TARZAC/Computer Products, Box 10203, Norfolk, Va. 23513, is on disk and is priced at $55.00.

CLUBS AND PUBLICATIONS

Clubs, user groups and organizations: We are offering special subscription rates to clubs, user groups and institutions. Minimum participation number is ten. Write (on your club/organization letterhead, please), for details concerning this special discount.

A new TRS-80 club has formed in the Portland, Or. area. A newsletter is planned. It meets at 7:00 p.m. first and third Thursdays of the month. Contact the president for exact location: John T. Warner, 13485 S.W. Driftwood, Beaverton, Or. 97005, phone (503) 646-2378.

Steve Johnston, Secretary of the TRS-80 Users Group of Spokane, Wa. tells us that their group has started a monthly newsletter. Also that their meetings are being held on the 2nd Wednesday of each month at Lincoln Mutual Savings, W. 818 Riverside in Spokane at 7:30 p.m. They already have a start at establishing a large software library. Contact Steve at S. 3718 Manito Blvd., Spokane, Wa. 992003.

AMRAD (Amateur Radio Research & Development Corp.) produces a newsletter covering Amateur Radio/Computer related material. Annual

(Continued on page 7)

If the purpose of life is to create new and challenging paths for men to explore and experience, then the computer has done its share...

Tell them all you saw it in the 80-U.S. JOURNAL!

80-U.S. JOURNAL

Publisher/Editor
I. Mike Schmidt
Editorial Assistant
Kathy Huston
Technical Editor
Tom Rosenbaum
Associate Editor
T.R. Dettman
Administrative Assistant
N. Marostica
Production Advisor
Rich Mulkey
Resource-In-Learning Consultant
G. B. Livingston, PhD.
Software Development
R. Groth
Reviewer
George Blank
Exectutive Secretary
Cathy Shappae
Layout and Design
Frontier Publishing
Printer
Peninsula Gateway

The **80-U.S. Journal** is published bi-monthly by 80-NW Publishing, 3110 North 31st Street, Tacoma, Washington, 98407. Phone: 206-759-9642. Subscription rates in United States : $16.00/1 year, $31.00/2 years, $45.00/3 years; Canada $20.00/1 year, $39.00/2 years, $55.00/3 years (sent First Class Mail); others $24.00/1 year, $47.00/2 years, $68.00/3 years all Must be remitted in U.S. funds. Printed in The United States of America. Bulk rate paid at Tacoma, Washington, 98407.

Editorial Remarks * *

It is our guess that most TRS-80 owners would not have a computer had it not been for the TRS-80. Apparently, the Tandy Corporation provided the "shoehorn" which allowed many undecided people to "ease into" a computer system of their own. The "pinch", although still there, is not as apparent as with some other systems and has the added advantage of not being as big. It may have been very clever design, or it may have been lucky default, but it is a fact that more people have TRS-80's than any other single system.

There are the complaints that some part or other of the system does not work as well as it should. There are problems with software and documentation. Some are justified, others must be taken with a grain of salt. In the final analysis, we will get out of our computer just about what we put into it.

There is an old story about the computer operator who complained about too much intervention with the machine. The manufacturer then brought out a new model, in a shiny steel case, with just two buttons and one indicator on it. The two buttons were marked "On" and "Off". A monkey was hired to push those two buttons, and the indicator came on periodically to tell the operator when to feed the monkey.

This may be driving the point too far, but points out the fact that we really don't want to get too automatic. After all, the trip there is usually more fun than the destination. Enjoy your problems and learn from them, there is no telling how far you can go with an open mind and a positive attitude!

LETTERS

CHARACTERS QUESTIONED

Gentlemen,

Thank you for your sample copy of 80-U.S. I was impressed with the very professional issue and its content and have entered my subscription.

Would appreciate a copy of your issue which described the hardware and software fix you used for lower/upper case. Have seen a few articles, but it is difficult to see which combination works best. Obviously yours works fine.

An additional question concerning changing characters, line numbers, and commands in a long program. Particularly at the upper end of RAM - the top 16 of 32K. have any of your readers had this problem or obtained a fix?

Roger L. Conarty
Albuquerque, NM

(There are several lower case mods that we are aware of. Some require that you hold the shift key down to type lower case (barbaric!). Ours does not, but-- ours does not support the Electric Pencil. Also, ours gives control characters when using TBUG. We are presently working up a mod which will give the best of both worlds, be easy to install and simple to use. Look for it in a future issue.

As for the other problem you mention: Yes, this seems to be a widespread complaint. It can be cause by using Brand X chips in the expansion interface, or having a marginal chip there. The RAM test will not always pick these chips out. It can also be caused by a mis-adjusted 5 volt power supply in the expansion interface. The BOM Active Cable Fix (supplied and installed by Radio Shack) goes a long way toward fixing most of these problems. Ask your local dealer about having it installed. Ed.)

PUNCH IT!

I got the second issue of 80-U.S. today. It's a FB *(fine business - Ham terminology, Ed)* magazine. I think the only improvement that could and should be made to it is to punch it so it could be kept in a three ring binder.

A comment to Philip Litchfield (Hangups, 80-U.S., Mar-Apr 79): I am using a TRS-80 Level II 16K with a phase lock loop converter. I send and receive CW 5 to 100 WPM, but don't go over 25. I send and receive RTTY on 2 Meter FM using AFSK. I have no problems with CW or RTTY

Frank M. Evans WA8KXB
Canton, OH

(We hope to be so big, so soon, that punching holes will be impractical - thanks for the comments Frank. Ed.)

COMPUTER ZIPS ZAPPED

Editor,

The period which Radio Shack provides (on the 779 line printer) just does not do the job. Is there anyone who can provide a fix to substitute two dots side by side? Possibly Radio Shack can be convinced of the necessity of this mod for those who want it.

I have seen an ad which offered to tell me about the one great error Radio Shack made in the Level II Basic System. I thought they had made several errors. In any case, does anyone know what this one great error is?

Do you remember years ago when the Post Office ran into the problem of not being able to read computer Zip Codes? Well, I recently mailed a few computer addressed things out and about 15% of them went to the strangest places. It appears that some of those old machines that confused the (cancelled zero) and the "eight" are around in smaller towns. If you have something really important put the Zip Code on by hand or use the "Oh".

Your magazine continues to be a creative and informative thing. Keep the pressure on and you will continue to keep the point position in this field.

Karl W. Joffs
Salt Lake City, UT

IT'S GREAT, BUT...

Mike,

The TEXT80 program is really great (80-U.S. Jan-FEb 79) however, I did find it necessary to make a few changes to make it work right when using the 62 character option. The first problem came when I left a blank line. it would cause an Illegal Function Call in line 560. This is caused by the For Next K loop in line 550. When a blank line is left the program tries to loop 62 times. On the 42nd loop the error is generated. This is because the value for Y is incremented by six on each loop. On the 42nd loop the value of Y is 258. As you can see, then line 560 asks the computer to find a blank space starting at character 258 of A$. This is impossible, and the error is generated.

It turns out that the maximum useful number of loops is 11 because at that point Y equals 66, a number which is larger than the possible number of characters in any line. To correct the problem a change to line 540 was made to change the value of W. Since W determines the number of loops it should be limited to 11. The re-write follows:

540 W=QZ-LEN(A$) :IF W > 11 then W=11

Gerry Sturgeon
Lynchburg, Va

(Or, type 3 spaces and ENTER for a blank line. Ed)

WORTHWHILE TRIVIA

Mike & Crew,

In the Jan-Feb 79 issue of 80-U.S., you ran an article on the un-newing of a program in Disk Basic. I responded, saying that I had not solved the problem with Level II. I have been working on it, and have met with success!

The main problem was that if a variable was used in the program it would be destroyed. What was needed was an adjustment of the memory size. This is accomplished by changing address 16633 thru 16636. Address 16633, 16634 contain the address of the end of the program, and address 16635, 16636 hold the end address of program and variables combined. Merely by poking these addresses with the proper codes, in addition to your suggestion, will restore the program to its original, untouched beatuy.

Another piece of trivia: The memory-related addresses in addition to the two pairs above, are 16637, 16561, 16598 and their mates (ie., 16638 etc.). 16637-8 hold the same address at 16635-6, and they must always hold the same address! The 16635-6 pair is used for memory calculations, and 16637-8 is used for calculating FRE(X$). 16561-2 hold the end address of the free string space. 16598-9 is the "base number", which hold the end address of the memory in your machine. This number never changes (unless you add memory).

Level II calculates memory using the following equations:
FRE(X$) = (PEEK(16561) + PEEK(16562)* 256)-(PEEK(16637) + PEEK(16638)*256)-
MEM, and
MEM = (PEEK(16544) + PEEK(16545)* 256)-(PEEK(16635) + PEEK(16636)*256) -14. (I guess the -14 allows for stack space). Thus, by POKEing in different values in the locations specified, you can add more free string space, memory size protect, etc., with little or no damage to the resident program and variables.

I still think...no, know that 80-U.S. is *the*

TRS-80 Publication! Keep on Computing!

Joe Sewell
PTU (Permanent TRS-80 User)
Wittman, Md

(In a near future issue, we will be publishing a short program based somewhat on the above information which will allow you to completely reclaim a lost program-especially helpful when your disk "Re-boots" back to DOS. Ed.)

SUGGESTIONS PLEASE

Dear Sir,

At the present time I own (or it owns me), a 16K Level II with 16K Expansion Interface and Disk. I have developed a business inventory system, which is good for the home, and a finance program which I use. My experience in programming is only seven months old.

What I would like to know is: 1) Can your Journal provide me with programming ideas so I can develop programs for home and business? 2) Useful hints and ideas on how to better my programming skill? 3) How can I get my wife interested in the dag-blasted computer?

Carl W. Mackey
Norfolk, Va

(You are apparently not doing too bad for only seven months of programming experience, Carl. We certainly hope you can find the answers in the pages of the JOURNAL, we are trying our best to put them there. As to getting your wife interested: 1) Do not take the computer to bed with you! and 2) Get her a Level 1 of her own (after she is hooked, you would never get to yours again) and 3) Let her develop a program to determine if your finances allow you to take her out to dinner at least once per week. You may go broke, but look at the fun you will have! Ed.)

COMPLAINT/COMPLIMENT

Editor,

I have thoroughly enjoyed both copies of 80-U.S. I have received so far. The JOURNAL is an excellent effort well worth the subscription price. However, I have two major complaints to make:

1. The AUTO feature of Level II works very well. Why then must you provide the programme listings with many line numbers that do not share a common increment?? There are many renumber programmes available, and one pass through before printing would save L2 users lots of needless typing.

2. The article on Modems (Mar-Apr 79 80-U.S.) was helpful, but didn't really address the proper issues for those seriously considering purchase. It would have been most helpful to know some extra details. For example, most true terminals and most computers use CONTROL (CTRL) and Escape (ESC) sequences for communication control (the DEC-10 uses CTRL-C for aborting a job step). It would be nice to know how this is handled. The same goes for the Duplex mode. If the RS-232C and terminal programme provided only allow half-duplex, then the TRS-80 would have problems communicating with most true timesharing systems. These two issues are of particular concern, since in most cases it's impossible to try out the package without buying it first!

I look forward to the future articles on TRS-80 FORTRAN. For those folk who do not know Fortran (or are rusty), I would strongly recommend the text "ELEMENTS OF FORTRAN IV", by Dr. H.C. Bezner, published by Goodyear Publishing (ISBN 0-87620-270-9). It is one of the very few texts that describes the 1966 ANSI Fortran Standard, and ignores the extended features most other texts love. Since TRS-80 Fortran is a standard Fortran with very few extended features, the text is a perfect learning tool. The text is under $10.00, a big bonus.

Rich Chambers
Waterloo, Onatrio, Canada

(We are going to gang up on you, Rich! We will answer the first and let the author of MODEM in your FUTURE, Terry, take the second. First, for renumber: We finally have obtained one that really works (without reading disk to tape, loading back in, renumbering and saving back on disk) and you will see (in this issue) that we are using what you have suggested. We found the Radio-Shack renumber very awkward and time consuming to use. Thanks for your comments. Ed.

Author replies -

Your comments are very well taken, and the information you ask for is naturally of interest. Each capability you desire is provided for by the Radio-Shack RS232C and the telephone interface. For example, the code for control is entered from the keyboard by holding down the shift and the down arrow keys together. Other CTRL codes can then be entered, such as CTRL H (backspace), CTRL B (escape). If you need other special codes, the RS232C manual tells how to modify the TERM program using TBUG to enter the desired codes in memory locatons 50B9H through 50BCH. In fact, TERM is located in memory at 5000H through 50BCH so that TBUG can be resident in memory at the same time.

The terminals (both the Type I and II also provide both half and full duplex operation. The Type II terminal is more compact and has the capability to answer as well as originate calls. This is worthwhile if you have someone to talk to who has an originate only modem. Terry)

NOTES ON BASIC

When you are pressed for space in memory, and need to cut bytes, it would seem natural to change the REM's to apostrophes. Don't! The REM takes three bytes on the screen, but only takes one byte in memory. The apostrophe, on the other hand, takes just one byte on the screen, but three in memory.

(Continued from page 5)

membership dues are $10.00. Write to Gerald Adkins, 1206 Livingston St. N., Arlington, Va. 22205

.....

Nick Sharp edits the Ventura County TRS-80 Computer Club Newsletter. The group numbers about 50 and meets the first Tuesday of each month at the Camarillo Public Library, 3100 Ponderosa Drive, Camarillo, Ca. at 7:00 p.m. Membership dues are $10.00 per year. Contact the club secretary, Lee Steinmetz at 567 West Loop Dr., Camarillo, Ca. 93030, phone (805) 484-1724.

.....

The first informal meeting of the Wichita Valley TRS-80 Users Group met on the evening of Feb. 27 in the recreation room of Perkins at Sikes Senter Mall in Wichita Falls, Tx. Meetings will be held the third Thursday of each month at 7:00 p.m. One of the first things they did was to publish a newsletter. For more information contact J. Wesley B.Taylor, PO Box 4391, Wichita Falls, Tx. 76308.

Troubleshooting Hardware

By M. Schmidt

When something in your system goes up in smoke it usually isn't too difficult to find the source of the smoke and repair whatever burned up. Even then, the component which did the burning is not always the culprit (an effect, rather than a cause).

This will become clear to you when you replace the burned component and it burns up again. Now you know it is an effect, not a cause, and you start looking for a reason.

It is at this point that a good idea of just how the circuit is supposed to function is important. Without some knowledge of the circuit you may well be lost. Since the symptom is well defined (a puff of smoke), you now need to find out why, but the "why" of it does not show itself as nicely as the effect did.

With a knowledge of the circuit (what it is supposed to be doing), a meter or osciloscope and a little effort, you can usually find the reason and correct it.

The real problems in troubleshooting hardware are the intermittent ones. Sometimes these are referred to as "glitches" (the name may well be a nice version of the same word with the g and l replaced by a b).

These little devils will rear their ugly heads, cause a system crash, and then return to a state of complete normalcy. Many times they will not even seem to be timed with any particular function, but will appear at random. Naturally, when you get out the scope to start looking for them, they quietly disappear.

There are a few ways to try and find intermittents. One is to observe the symptoms they cause, then start to replace every component remotely connected with that kind of function. This is known as "shotgunning" or "Easter egging" the problem. Technicians usually hate to admit they are using this approach.

Another way to find intermittents is to "smoke them out". This involves raising the supply voltage (or the temperature) of the circuit in hopes of making an intermittent go "solid". It is easier to find a component which never works than one which sometimes doesn't work.

The application of heat, usually by holding a soldering iron close to the suspect component, will sometimes cause a marginal component to fail completely. The trick is to be heating the right component. This is just a ~~one~~ respectable form of "Easter egging", but it has been known to work.

The most difficult of problems is the one where there is an apparent glitch causing a crash, but it turns out to have unrelated causes, all giving the same symptom.

This is where you can find a good cause and repair it, only to find the symptom is still there. You begin to have doubts about your own sanity, especially after finding and repairing as many as three causes, only to see the symptom come back.

This is precisely why a fix (such as the R-S BOM Active Cable Fix) will not always cure a symptom. In this case the symptom is that the disk re-boots to DOS in the middle of a program. One owner reported that he replaced the capacators in the expansion interface with good quality tantalum capacators - and the problem went away. Others have had the "pregnant cable" installed and solved the problem, while others who had it installed still have the problem. Another user reports that his disk re-boots when the refrigerator kicks in, and that readjustment of the 5 volt power supply in the expansion interface solved the problem.

In my own case, none of the re-booting symptoms occured over a period of eight months. Then the RS 232 was installed and the symptoms appeared regularly. Since this was the only known change, the RS 232 was removed for a period of two weeks. Not one re-boot during that period. Putting the RS 232 back into the expansion interface caused rebooting again. There is a DC to DC converter on the RS 232 board which may be causing Radio-Frequency hash to run through the rest of the circuitry. Then again, it may not be that at all.

These examples serve to illustrate the point that it may be more than one cause, but just one symptom. It also illustrates the challenge presented to anyone who feels inclined to become an electronics technician.

More important than all the tools in your bag, is a good understanding of the circuits you are working on. There is no substitute for knowledge. And that, coupled with a little imagination, can help you through even the worst intermittent glitches.

THE TRS-80 ASSEMBLY LANGUAGE DEVELOPMENT SYSTEM.

(A STEAL AT TWICE THE PRICE)

A short time ago, Microsoft introduced TRS-80 FORTRAN— a complete ANSI-standard FORTRAN with macro assembler, linking loader, and text editor, all for only $350. **The response has been overwhelming.**

Many TRS-80 users even told us, "The assembly language development software alone is worth that price." We think they're right, of course, but we've made it an even better deal.

ANNOUNCING: THE TRS-80 ASSEMBLY LANGUAGE DEVELOPMENT SYSTEM FOR $175.

For half the price of the TRS-80 FORTRAN Package, you can buy the TRS-80 ASSEMBLY LANGUAGE DEVELOPMENT SYSTEM, including:

EDIT-80 A fast, random access text editor that's easy to use and loaded with features. Lets you insert, replace, print or delete lines; edit individual lines; renumber lines in a file; and find or substitute text.

MACRO-80 The best Z80 assembler anywhere. MACRO-80 supports a complete Intel-standard macro facility plus many other "big computer" assembler features: comment blocks, octal or hex listings, 8080 mode, titles and subtitles, variable input radix (base 2 to base 16), and a complete set of listing controls.

LINK-80 Loads your relocatable assembly language modules for execution and automatically resolves external references between modules.

CREF-80 Gives you a complete dictionary of program symbols, showing where each is defined and referenced.

The Microsoft TRS-80 FORTRAN Package is still available for $350. Or, for HALF PRICE, get the TRS-80 ASSEMBLY LANGUAGE DEVELOPMENT SYSTEM.

Either way, it's a steal.

TO: Microsoft, 10800 N.E. 8th, Suite 819, Bellevue, WA 98004

☐ Send me the works! TRS-80 FORTRAN Package for $350.

☐ Send me half the works! TRS-80 Assembly Language Development System for $175.

☐ Check enclosed ☐ Master Charge ☐ VISA

CARD NUMBER EXP. DATE

CARDHOLDER'S SIGNATURE

NAME

ADDRESS

CITY STATE ZIP

MICROSOFT

Dealer Inquiries Invited

STRING PACKING TECHNIQUES EXPOSED!

By M. Schmidt

Last issue, we presented four new programs which all included "String Packing". This is a technique in which string elements, i.e., CHR$(130) or 191, may be "packed" into memory using only one byte of space instead of anywhere from 3 to 9 bytes. We also mentioned using a command other than USR(0) to vector to a machine language routine.

In this issue we will discuss string packing, and in the following issue will cover the vector problem.

The general idea of string packing is as follows: A string variable is set up as a line statement enclosed in quotes, with as many empty spaces between the quotes as there are string elements to be inserted. Example:

```
10 A$="                    "
```

The actual string elements are then set up in a DATA statement. A subroutine is set up (using high program line numbers) which uses a FOR-NEXT loop to poke the string elements into the A$ string. There should be as many elements as there are blank spaces in A$ (between quotes). Here is an example you can try: Make line 10 above, with A$ equal to ten spaces between quotes. Then type in the following program:

```
30 GOSUB99
40 STOP
99 RESTORE
100  DATA 191, 191, 191, 198, 185, 184, 183, 182, 181,
   180
110 X=PEEK(VARPTR(A$)+2)*256+PEEK(VARPTR(A$)+1)
120 FOR I=1 to 10
130 READ J
140 POKE X+I-1,J
150 NEXT I
160 RETURN
```

Prior to running this program, A$ was equal to ten spaces between quotes. After the program is run, A$ is equal to ten command words:

```
A$="USINGUSINGUSINGPOINTCLOADCLEARAUTODELET
ELLISTLIST"
```

A check of memory size, using this and the normal method, will show that (in this case) only one-third the memory space for the string is used. If the string elements were CHR$(XXX), then as many as eight bytes per element can be saved.

Machine language can be packed in the same way. It is only necessary to run each string variable (such as A$ used above) through the subroutine, changing the variable name in line 110 and the number of elements in 120. After all the variables in a program are thus packed, the subroutine may be deleted. The string elements themselves are never put into the variable in the program, rather they are entered into line 100 as data elements.

A by-product of this technique is that the resulting program listing is very difficult to read. Also, it causes the screen to roll and will drive a printer crazy. This is not very desireable in most cases, (how do you get a decent looking printout?), but it does afford a limited degree of protection to your proprietary methods.

It was not our intention to confuse the issue, but the memory space saved makes this technique worth the effort, especially in programs using string graphics.

CONTENTS

Volume II, Number 3
May/June 1979

80-U.S.

The TRS80 Users Journal

FEATURES

DEPARTMENTS

COLUMNS

software review

STARFLEET ORION

By G. Blank, Leechburg, PA

This is the most impressively packaged computer game I have yet seen. It comes with a cassette tape, a tablet of worksheets, and two attractive blue booklets, one containing the basic rules for the game, and the other containing a catalogue of variations. Some idea of my first impression can be stated simply by reporting that it took $2.25 in postage to mail it to me.

Starfleet Orion is better described as a game system rather than a game. You can play one of the 12 games listed in the book of scenarios or create your own games, providing for battles between up to 18 spaceships or 17 ships and one planet. Though set up for two sides, it is also possible to have several players operating independently. A one player version is planned for the near future.

There are three programs on the tape, named ORION, BUILDER, and INTRO. ORION is the main program, which manages the science fiction tactical battle between the space ships. INTRO is the data for the first scenario, a battle between the missle frigate Britomartis and the torpedo frigate Bellatrix V. As a simple game to introduce new players to construct data tapes for other scenarios, either the twelve scenarios in the book or your own. You can define spaceships by specifying power for energy, drive, attack beam, shields, and the thickness of armor, number of launch tubes, missiles, and torpedoes. You can either choose from 22 spaceship types listed in a chart, from Battleships to Transports, or design your own. You can also specify damage to any of the ships to simulate a crippled ship being escorted back to base or a scenario of your own.

The twelve scenarios are listed in order of complexity from INTRO to ARMAGEDDON, a battle involving the planet Autarchia, two battleships, two pocket battleships, 8 cruisers of three different types, and 3 fighters, taking 6 hours to play.

In the actual play of the game, each ship is represented as a single square on the screen, with an identifying code next to it, the numbers 1 to 9 for the ships of one side and A through I for the other. If there is a planet, it is labeled P. On each turn you may choose to move each ship individually, use a tractor beam to move another ship, put energy into your shields, or fire your three weapons; beam, torpedoes, and missiles.

The game is challenging and complex, and there are so many variations that it can be fun for a long time. Strategy and tactics offer so many possibilities that a skillful player will be one who has played a lot of games. Yet it is possible to adjust the power of each side to give the weakest player a chance against the strongest.

I feel the game has only two serious drawbacks. The graphics display, already described, is quite dull, and I hope a future version will show different ships in picture form. The second drawback is that when there are a lot of ships, it can take a terribly long time to enter all the moves for each turn.

My ratings:

Instructions: Moderately difficult
Documentation: Excellent
Challenge: Very Good
Graphics: Poor
Pacing: Fair
Recommendation: Worth the price if you are looking for a challenging two person battle game.

Starfleet Orion is available for a 16K Level II TRS-80 or an 8K Pet from Automated Simulations, Dept. 8U, P.O. Box 4232, Mountain View, CA. 94040. The price $16.95, and California residents must pay sales tax.

IF YOU OWN A TRS 80 *.....

TRS-80 is a product of Radio Shack, A Tandy Corporation

SELECTRA-PRINT PUTS YOU JUST A CABLE LENGTH AWAY FROM A "HARD—COPY" (Typewriter Quality: clean, clear, high-fidelity) Word Processing System!

. . .just a simple hook-up with the supplied cables and your *SELECTRA-PRINT* is ready to run.

For Word Processing on a TRS-80, for example, just command "LPRINT" and *SELECTRA-PRINT* automatically outputs clear, clean, high-fidelity, hard copy. . . . and of course you can use it to print-out any other information you might need.

SPECIAL NOTE:
SELECTRA-PRINT is a Selectric II typewriter and although it has been modified for computer print-out, it may still be used as a standard office typewriter.

PRICE: $1,850.00* *TRS-80 Version $1925.00

OPTIONS:

Dual Pitch	$125.00
Correction Feature	$125.00
Noise Reduction Feature	$50.00
RS-232 Interface	$195.00

Direct International Sales:
International Sales Division
17648 Orna Drive
Granada Hills, CA 91344, USA

Discounts Available to
EDUCATIONAL ACCOUNTS
Contact **Delores Sun P.O. Box 8394**
***Ann Arbor, MI 48105* (313) 665-8514**

* ***SELECTRA-PRINT is VERSATILE!***
SELECTRA-PRINT is compatible with most micro- and mini-computers including ***PET*** ● ***Apple*** ● ***Heath H8*** ● ***IMSAI*** ● ***Cromemco*** ● ***Alpha Micro-systems*** ● ***Space Byte*** ● ***North Star Horizon*** ● ***SWTP*** ● ***Vector Graphic*** ● ***Sol*** ● ***Polymorphic*** ● ***Digital Group*** ● ***Ohio Scientific*** ● ***Altair*** ● ***Sorcerer*** ***Xitan*** ● ***Rex*** ● ***KIM*** ● ***EXORcisor*** . . . .

***"From now on,* PRINT-OUT takes on a new meaning"**

M30
micro computer devices inc.

960 E. Orangethorpe, Bldg. F
Anaheim, California 92801
Telephone (714) 992-2270

"Innovators to the Microcomputer Industry"

COMPUTER SYSTEMS CENTER

226½ West Foothill Blvd.
Claremont, CA. 91711

714/621-5059

TRS-80 TM Mods

1 Upper/Lower Case
installed with switch between "jibberish" and standard Level II. With software for Level II to turn "jibberish" into properly displayed data.

2 Level I/II
installed in same machine switch selected.

3 Clock Rate
switch selected 1.77MHZ (standard) or 2.66 MHZ (fast) cut your processing time by 1/3.

TRS-80 TM Components

4 Serial Interface
installed in keyboard. With software driver for Level II

5 16 K RAM
installed & tested

6 16KRAM
chips only (you install)

7 Floppy Disc
Pertec FD-400 with cable, enclosure & power supply. We will test on your expansion interface

8 Integral Data Printer
pressure feed platten, upper/lower case, programmable character size, 50 c.p.s. continuous impact printer.

9 Centronics Up-1
as item 8 but using silvered paper (see in R.S. catalog).

10 12" B&W TV
modified for TRS-80. Switchable as TV or video monitor. Can be used with B&W camera.

Package units properly. We are not responsible for shipping damage.

TRS80 System L2 16K
Complete - $849.00

Item	Price	You send
1	49.00	keyboard
2	29.00	keyboard
3	29.00	keyboard
4	49.00	keyboard
5	119.00	keyboard or exp. intf.
6	99.00	nothing
7	399.00	exp. optional
8	849.00	exp. optional
9	449.00	exp. optional
10	149.00	nothing

80-U.S. SOFTWARE

PRESENTS

ANIMATION PLUS SOUND!

1.NEW! ANDROID NIM with SOUND! The TRS-80's first animated graphics winner now has more animation and SOUND - by Leo Christopherson, Level II, 16K **$14.95**

1. NEW!LIFETWO with SOUND! Conways game of life at an astounding 100 generations per minute! PLUS Leo C's animated "talking" creatures playing the "BATTLE OF LIFE" in one 16K, Level II program **$14.95**

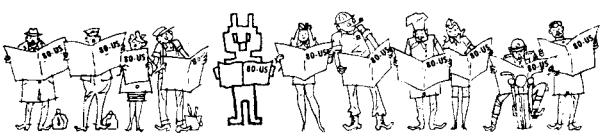
3. NEW!SNAKE EGG! with SOUND. This version is "21", with talking snakes, will keep you entrhalled for some time! by leo Christopherson **$14.95**

DEALER INQUIRIES INVITED

1. NEW! CUBES! Another Leo Christopherson special! Gives the solution to "Instant Insanity" ® or numbered blocks. Let your computer go crazy trying to figure how to arrange the blocks on your screen. Level II, 16K **$9.95**

2. SPECIAL! The original ANDROID NIM. While they last! The first TRS-80 animated graphics game **only $8.00**

3. TEXT80 A simple Basic Text Editor with no complicated sub-commands you need to learn. On a formatted disk (or customer supplied DOS disk) **$12.00**



YOU'VE READ THEM ALL- NOW GET THE BEST!

The 80-US Journal

A PUBLICATION FOR TRS-80 USERS - COVERS **ALL** TRS-80 CAPABILITY!

206/759-9642

Master charge/Visa accepted

MAIL TO:
80-U.S. Journal
P.O. Box 7112
Tacoma, WA. 98407

80-U.S. SOFTWARE

Please order by the name of the program

NAME **AMOUNT**

.....

.....

.....

NAME _____

ADDRESS _____

CITY, STATE, ZIP _____

MAIL TO: **80-U.S. SUBSCRIPTION ORDER FORM**

80-U.S. Journal
P.O. Box 7112
Tacoma, Wa. 98407

SUBSCRIPTION RATES:

U.S.	CANADA
$16.00/1 year	$20.00/1 year
$31.00/2 years	$39.00/2 years
$45.00/3 years	$55.00/3 years

NAME _____

ADDRESS _____

CITY, STATE, ZIP _____

your thinking pattern if the test is taken on the basis of answering with your first impression.

For more information on intuition and intellect, you may send a self addressed stamped envelope with $1.00 to:

Dr. G. B. Livingston
4401 Sunset Beach Road West
Tacoma, WA. 98466

This will also put on the mailing list for future developments.

.....

IF YOU ARE ANALYTIC:

You perceive the world through your senses and code that information in terms of language. You do best with conceptual rather than visual similarities. You perceive in detail, record in memory, and consciously process the information.

You like applying systematic procedures and formulae to solve problems and work well with standarized procedures. You enjoy using the skills you have developed and can be relied upon to work through assigned problems. Your boss undoubtedly appreciates your responsibility towards your work and your reliability, when it comes to assigned tasks.

You tend to stay in the same field for long periods of time, developing proficiency with the tools of your trade. If you responded to bigger and better opportunities, it is likely that they came to you rather than the other way around. A change in careers will most likely capitalize on the skills you have already acquired.

You tend to be "irked" by required changes and unexpected "monkey wrenches" in your routine. In general you are considered stable, patient, and live easily and comfortably in your own environment. If you own a TRS80, it is likely that you use it in business rather than it being an avocation or a hobby.

Your major strength is your use of reasoning and logic to solve problems. You have the ability to do complicated high order thinging processes such as synthesis, extrapolation, and evaluation. If you haven't capitalized on this ability, it is probably due to the lack of motivation. It may be useful to have an intuitive person about to inspire you. If you keep company with intuitives, you can pick their brains for new and exciting ideas. It may require patience to put up with intuitive thinking patterns, but it will benefit you in the long run.

Your goal setting and implementation requires constant effort. you do best by writing down your goals and reviewing them daily to measure your progress against them.

Your memory potential is good and will improve as its use becomes more essential to your success. You will use it as a tool box to provide information as needed rather than to develop new ideas.

You are likey to enjoy socializing, to have many friends, and to enjoy yourself at large organized parties. You like regularly scheduled functions and look forward to socializing.

Unless there have been many negative influences in your life, you tend to live easily and comfortably in your environment. You are patient and tolerant, and have well formulated and stable attitudes, values and morals. You schedule well in advance and tend to stick to the schedule.

It is likely that you adjusted well and enjoyed elementary and secondary school. It is likely that you were still doing well when you left school.

IF YOU ARE INTUITIVE:

You perceive in sensory images. You have an innate space-time rhythm which can make you excel in physical skills and/or artistic expression (particularly music). You tend to be imaginative and creative in your thinking and innovative in behavior.

You have a zest for tackling complex and challenging problems, and a drive for diversity that could lead you through several career changes. Others may view this as restlessness and instability. You get inspired easily and often have abundant energy, enthusiasm, and patience to work through complexity. If you do not understand your intuitive nature and learn how to capitalize on it, you are vulnerable to restlessness, impatience, and overstress. You could easily "turn off" or become a "wheel spinner"--indecisive, vacillating, and ineffective.

Goal implementation strategies that

are most effective for you are those that plant an idea and let it "incubate" and then grab the solution when it surfaces. You find that you often find solutions to your problems in your sleep, and would be well advised to keep a notepad by the bed to jot down the solutions as they occur.

You have probably surpassed those who did better than you school. Your memory is likely to be used to process concepts rather than details. Logic and higher thinking skills occur only in your out-of-consciousness time. Logical thinking at the conscious level can be a benefit to you but you may need to make a conscious effort to do it.

Prolific "input" enchances intuitive thinking. Reading voraciously, reflection, and thinking in states of altered awareness as well as communication through writing, serves to sharpen those skills you are best at.

You can usually go directly to the solution of any complex problem after an "incubation" period where your subconscious processes the data and arrives at a solution. You are rarely aware of the intermediate steps in this processing and in fact it tends to be more effective if you involve yourself in something else while it is going on. It is also effective to go into a relaxed, altered state of consciousness such as mediation while processing●

Program Notes:

Lines	Comments
10-60	sets up initialization + first screen
70-100	sets up for printed output if desired checks memory location 14312 to see if the printer is ready and prints heading
100-120	goes to subroutine 630 if instructions are needed
130-140	initializes the array of cutoff values for separating the types of responses
160-340	loops over twenty questions
170-260	formats the screen display
270	checks memory location 14400 to see if a left or right arrow has been entered
270-310	loops continuously through checking which arrow is pressed, moving the marker in response and stopping when the enter key is pressed (line 320)
300	'human engineering', if the marker goes outside the allowable range, the display is reset
330	record answer to question
350-370	compute mean and average deviation statistics
380	if printing is desired, then go to subroutine 1740
390-410	select the proper response and go there
430-620	adjectives for the test
630-760	instructions for the test
770-1640	responses for the test
1660	if printing, this advances the paper to where it can be torn off easily
1690-1730	error trap - any program errors will go here instead of returning to the READY condition.
1740-1800	sets up to use the line printer with PRINT statements by putting the address for the line printer driver in place of that for the screen
1800-1820	restores normal print operation

```
10 REM 80-U.S.JOURNAL-PERSONALITY PROFILE-G B LIVINSTON, PHD
20 REM PROGRAM BY CONSULTING SERIVES NW (C)1979 80-NW
       PUBLISHING CO
30 CLEAR1000:ON ERROR GOTO 1690
40 DIMSC(20),TV(6):RANDOM
50 CLS:PRINTTAB(15)" PERSONALITY PROFILE":PRINT
60 INPUT"WHAT'S YOUR NAME? ";NM$
70 INPUT"DO YOU WANT PRINTED OUTPUT";Y$
80 IFLEFT$(Y$,1)="Y"THENPRT=1ELSEPRT=0
90 IF(PRT=1)*(PEEK(14312)>127)THENGOSUB1770
100 IF(PRT=1)THENLPRINTSTRING$(10,138):LPRINT"ANALYSIS FOR ";
     NM$:LPRINTCHR$(138)
110 Y$=" ":INPUT"DO YOU WANT INSTRUCTIONS";Y$
120 IFLEFT$(Y$,1)="Y",GOSUB630
130 FORI=1TO6:READTV(I):NEXT
140 DATA30,38,42,45,50,55
150 A$="MOVE THE 'X' WITH THE RIGHT/LEFT ARROW KEYS UNTIL IT
       INDICATES WHICH WORD(S) YOU PREFER MOST, MOVE
       FARTHER FOR A STRONGER PREFERENCE."
160 FORI=1TO20
170 CLS:PRINT@0," PERSONALITY INDEX";
180 PRINT@64 ,"A TEST TO DETERMINE RIGHT/LEFT BRAIN PERFERENCE"
190 PRINT@192,"QUESTION ";I;
200 RR=RND(2):IFRR=1THENREADB$,C$,D$,E$ELSEREADD$,E$,B$,C$
210 PRINT@320,A$;
220 PRINT@513,B$;:L=LEN(D$):PRINT@573-L,D$;
230 PRINT@577,C$;:L=LEN(E$):PRINT@637-L,E$;
240 PRINT@705,STRING$(61,46);
250 IFRR=1THENX=34ELSEX=26
260 PRINT@705+X,"X"
```

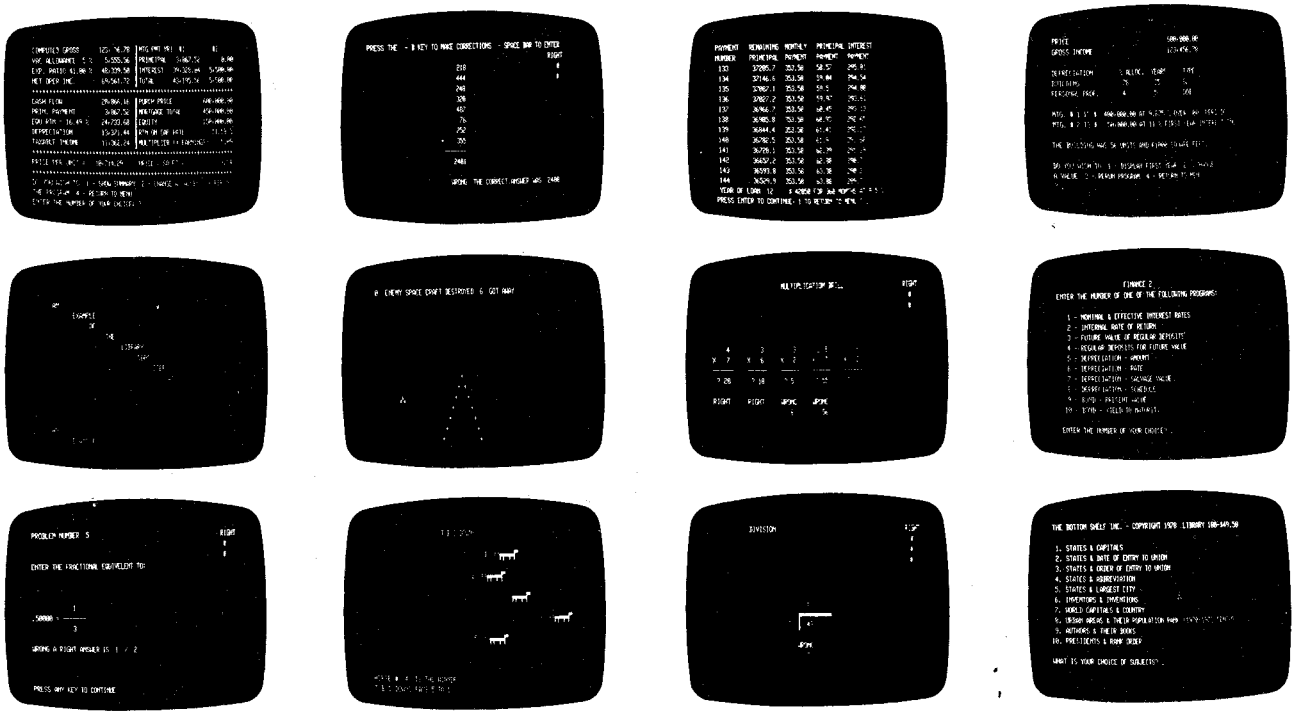
```
270 Z=PEEK(14400)
280 IFZ=32THENX=X-1:PRINT@706+X,".":PRINT@705+X,"X"
290 IFZ=64THENX=X+1:PRINT@704+X,".":PRINT@705+X,"X"
300 IF(X<0)+(X>60)THENPRINT@832,"ANSWER OUT OF RANGE, TRY
     AGAIN";:FORJ=1TO500:NEXTJ:PRINT@832,STRING$(40,32);:
     PRINT@704," ":PRINT@766," ":GOTO250
310 Z$=INKEY$:IFZ$=""THEN270
320 IFASC(Z$)=13THEN330ELSE270
330 IFRR=1THENSC(I)=X+10ELSESC(I)=70-X
340 NEXTI
350 CLS:PRINT@468,"PROCESSING"
360 MN=0:FORI=1TO20:MN=MN+SC(I):NEXTI:MN=MN/20
370 DEV=0:FORI=1TO20:DEV=DEV+ABS(MN-SC(I)):NEXTI:DEV=DEV/20
380 IFPRT=1THENGOSUB1740
390 FORI=1TO6
400 IFMN<=TV(I)ONIGOTO770,870,950,1060,1350,1420
410 NEXT
420 GOTO1490
430 DATA ESTABLISHED,PERFORMANCE,INNOVATIVE,PERFORMANCE
440 DATA DELIBERATE,REASONING,IMMEDIATE,INSPIRATION
450 DATA ,SIGN,,SYMBOL
460 DATA ,PRESTIGE,,ORIGINALITY
470 DATA ,REASONABLE,,FASCINATING
480 DATA ,REASONABLE,,CREATIVE
490 DATA ,ORGANIZED,,INNOVATIVE
500 DATA ,STABILITY,,INGENUITY
510 DATA STABLE,FORMULA,CHALLENGING,THEORY
520 DATA ,CONSTRUCT,,DESIGN
530 DATA ,STABLE,,FLUID
540 DATA ,OBJECTIVE,,ABSTRACT
550 DATA COURSES,IN FACT,COURSES,IN THEORY
560 DATA PRECISE,MEANING,ORIGINAL,EXPRESSION
570 DATA ,TIME,,SPACE
580 DATA ,CERTAINITY,,THEORY
590 DATA ,FACTS,,IDEAS
600 DATA ,CONSISTENT,,INGENIOUS
610 DATA ,REALITIES,,POSSIBILITIES
620 DATA ,PRACTICAL,,INSIGHTFUL
630 REM INTRODUCTORY MATERIAL
640 CLS:PRINTTAB(15)"TAKING THE TEST"
650 PRINT"(1)  MOVE THE MARKER (X) ALONG THE LINE WITH THE LEFT AND"
660 PRINT"RIGHT ARROWS UNTIL IT GIVES A BALANCE BETWEEN HOW WELL"
670 PRINT"EACH WORD DESCRIBES YOU, THEN PRESS ENTER"
680 PRINT"(2)  IF YOU GO OFF THE LINE TO THE RIGHT OR LEFT, AN"
690 PRINT"ERROR MESSAGE WILL BE DISPLAYED, AND YOU WILL BE ASKED"
700 PRINT"TO TRY AGAIN"
710 PRINT"(3)  AFTER THE TWENTIETH QUESTION, THE SYSTEM WILL"
720 PRINT"AUTOMATICALLY BLANK OUT THE SCREEN AND COMPUTE YOUR"
730 PRINT"PERSONALITY TYPE BASED ON EXTENSIVE TESTING WITH STANDARD"
740 PRINT"GROUPS. A MESSAGE WILL THEN BE DISPLAYED TO TELL"
750 PRINT"THE BRAIN FUNCTIONING GROUP INTO WHICH YOU FALL"
760 PRINT@960,;:INPUT"PRESS ENTER TO START THE TEST";:RETURN
770 REM ANALYTICAL I
780 CLS:PRINTTAB(15)"ANALYSIS":PRINT
790 PRINT"YOU HAVE A VERY STRONG LEFT BRAIN PATTERN THAT"
```

```
800 PRINT"ALMOST DICTATES CAREERS THAT YOU WILL OR WILL"
810 PRINT"NOT HAVE COMPATIBILITY FOR,  AND DETERMINES HOW"
820 PRINT"YOU IMPLEMENT GOALS.":PRINT:PRINTTAB(30)"**OR
830 PRINT"YOU ARE A SNEAKY INTUITIVE WHO HAS ALREADY DISCOVERED"
840 PRINT"YOUR PATTERN AND WANTS TO KNOW WHAT THE OTHER"
850 PRINT"PEOPLE ARE LIKE."
860 GOSUB1810:PRINT@960,;:INPUT"PRESS ENTER WHEN READY TO
    CONTINUE";:GOTO1650
870 REM ANALYTICAL II
880 CLS:PRINTTAB(15)"ANALYSIS":PRINT
890 PRINT"YOU HAVE A STABLE SENSORY PERCEIVING AND LEFT"
900 PRINT"BRAIN PROCESSING PATTERN THAT HAS STRONG"
910 PRINT"IMPLICATIONS FOR CAREER COMPATIBILITY, "
920 PRINT"INTERPERSONAL RELATIONSHIPS, AND GOAL"
930 PRINT"IMPLEMENTATION STRATEGIES."
940 GOSUB1810:PRINT@960,;:INPUT"PRESS ENTER TO CONTINUE";:
    GOTO1650
950 REM ANALYTICAL III
960 CLS:PRINTTAB(15)"ANALYSIS":PRINT
970 PRINT"YOU HAVE A MILD BUT CONSISTENT AND DEFINITE"
980 PRINT"SENSORY PERCEPTION AND INTELLECTUAL PROCESSING"
990 PRINT"PATTERN.  LIMITED COMPUTER CAPACITY DOES NOT ALLOW"
1000 PRINT"THE ITEM ANALYSIS THAT WOULD DETERMINE WHETHER"
1010 PRINT"THIS IS DUE TO LESS PREFERENCE FOR LOGIC OR"
1020 PRINT"GREATER INCLINATION FOR INTUITIVE PROCESSING."
1030 PRINT"YOU PROBABLY HAVE GREATER MOTIVATION THAN THE"
1040 PRINT"AVERAGE PERSON OF THE ANALYTICAL TYPE."
1050 GOSUB1810:PRINT@960,;:INPUT"PRESS ENTER TO CONTINUE";:
     GOTO1650
1060 REM MIDDLE OF THE ROAD
1070 IF(MN=44)*(DEV=0)THENGOSUB1810:CLS:PRINTCHR$(23):
    PRINT@448," NO WORDS CAN DESCRIBE YOUR":PRINT"
    FASCINATION WITH THE 'ENTER' KEY":
    FORI=1TO2000:NEXTI:GOTO1650
1080 CLS:PRINTTAB(15)"ANALYSIS":PRINT
1090 PRINT"WOW!  THREE PATTERNS ARE POSSIBLE:"
1100 PRINT"(1)  YOU HAVE LOCATED THE STRONG RESPONSES FOR BOTH"
1110 PRINT"LEFT AND RIGHT BRAIN FUNCTIONING AND GIVEN BALANCED INPUT"
1120 PRINT"TO SEE WHAT THE READOUT WILL SAY.  YOU ARE TESTING THE TEST!"
1130 PRINT"TRY TESTING YOURSELF!!!!"
1140 IFPRT=1THEN1190
1150 FORI=1TO5000:NEXTI
1160 CLS:PRINTCHR$(23):PRINT@468,"** OR **"
1170 FORI=1TO2000:NEXT
1180 CLS:PRINTTAB(15)"ANALYSIS":PRINT
1190 PRINT"(2)  YOU HAVE DEVELOPED PROFICIENCY IN BOTH LEFT BRAIN"
1200 PRINT"AND RIGHT BRAIN PROCESSING AFTER CAREFUL SELF ANALYSIS AND"
1210 PRINT"HEAVY CONCIOUS EFFORT TO DEVELOP THE OPPOSITE PROCESSING"
1220 PRINT"PATTERN.  IF COMPETANCY FOLLOWS PREFERENCE, THIS LEAVES THE"
1230 PRINT"WORLD AND EVERYTHING IT HAS TO OFFER OPEN TO YOU.  YOU ARE"
1240 PRINT"ONE OF THE FEW PEOPLE TO ACCOMPLISH THIS AND YOU ARE BEYOND"
1250 PRINT"PATTERNS."
1260 IFPRT=1THEN1310
1270 FORI=1TO9900:NEXT
1280 CLS:PRINTCHR$(23):PRINT@468,"** OR **"
1290 FORI=1TO2000:NEXT
```

```
1300 CLS:PRINTTAB(15)"ANALYSIS":PRINT
1310 PRINT"(3)  YOUR NATURAL PREFERENCE HAS BEEN OVERRIDDEN BY PARENTAL"
1320 PRINT"OR SOCIETAL 'SHOULDS' AND YOU HAVE MADE HEAVY CONCIOUS"
1330 PRINT"EFFORT TO CONFORM.  THIS PATTERN CAN CAUSE DISTRESS"
1340 GOSUB1810:PRINT@960,;:INPUT"PRESS ENTER TO CONTINUE";:
    GOTO1650
1350 REM INTUITIVE I
1360 CLS:PRINTTAB(15)"ANALYSIS":PRINT
1370 PRINT"YOU HAVE A MILD BUT CONSISTENT AND DEFINITE"
1380 PRINT"INTUITIVE PATTERN.  THE DESCRIPTION WILL OVERSTATE"
1390 PRINT"YOUR CASE AND LEAVE A WIDER MARGIN OF ERROR"
1400 PRINT"BECAUSE YOU HAVE SOME STRONG LEFT BRAIN PATTERNS"
1410 GOSUB1810:PRINT@960,;:INPUT"PRESS ENTER TO CONTINUE";:
    GOTO1650
1420 REM INTUITIVE II
1430 CLS:PRINTTAB(15)"ANALYSIS":PRINT
1440 PRINT"YOU HAVE A MODERATELY STRONG INTUITIVE PATTERN"
1450 PRINT"AND THE DESCRIPTION SHOULD DESCRIBE YOU QUITE "
1460 PRINT"ACCURATELY EXCEPT FOR THE CONTAMINATING FACTORS"
1470 PRINT"WHICH ARE NOTED."
1480 GOSUB1810:PRINT@960,;:INPUT"PRESS ENTER TO CONTINUE";:
    GOTO1650
1490 REM INTUITIVE III
1500 CLS:PRINTTAB(15)"ANALYSIS":PRINT
1510 PRINT"THERE ARE TWO POSSIBILITIES:"
1520 PRINT"(1)  YOU HAVE AN EXTREMELY STRONG INTUITIVE PATTERN"
1530 PRINT"WITH ALL OF THE STRENGTHS AND VULNERABILITIES"
1540 PRINT"INHERENT IN THAT PATTERN. YOUR PREFERENCE IS SO"
1550 PRINT"STRONG THAT YOU PROBABLY PERFORM WITH HIGH"
1560 PRINT"COMPETENCE IN THE AREA OF EXPERTISE AND YOU ARE"
1570 PRINT"UNHAMPERED BY THE LIMITATIONS INHERENT IN YOUR"
1580 PRINT"PATTERN. PAIRING WITH AN INTELLECTUAL COULD BE OF"
1590 PRINT"VALUE TO YOU."
1600 PRINT"(2)  YOU HAVE PLAYFULLY AND CAPRICIOUSLY FOUND ALL OF THE"
1610 PRINT"STRONGEST INTUITIVE RESPONSES TO SEE WHAT THE PRINTOUT WILL"
1620 PRINT"SAY.  YOUR CUNNING IS REWARDED.    I SUSPECT YOU"
1630 PRINT"ALREADY KNOW YOUR OWN PATTERN."
1640 GOSUB1810:PRINT@960,;:INPUT"PRESS ENTER TO CONTINUE";:
1650 GOSUB1810:CLS:PRINT@960,;:INPUT"ANYONE ELSE INTERESTED";Y$
1660 IFPRT=1THENLPRINTSTRING$(15,138)
1670 IFLEFT$(Y$,1)="Y"THENRESTORE:GOTO50
1680 END
1690 REM ERROR TRAP
1700 CLS:PRINTCHR$(23):PRINT@448,"ERROR"
1710 PRINT@512,"PRESS ENTER TO RESTART TEST"
1720 INPUTZZ$:CLS
1730 RESUME150
1740 REM LINE PRINTING ROUTINE
1750 IF PEEK(14312)>127 THEN GOSUB1770
1760 POKE16414,141:POKE16415,5:RETURN
1770 CLS:PRINTCHR$(23)
1780 PRINT@448,"PRINTER NOT READY"
1790 PRINT@512,"PRESS ENTER WHEN READY"
1800 INPUTZZ$:CLS:RETURN
1810 REM RETURN FROM PRINTER
1820 POKE16414,88:POKE16415,4:RETURN
```

Level II TRS-80 + The LIBRARY 100

You shouldn't have one without the other!!



Finance: Present Value of a Future Sum—Simple Interest for Days—Future Value of a Present Sum—Amortization Schedule—Interest Rate: Compound Interest—Interest Rate: Installment Loan—Days Between Dates—Term of an Installment Loan—Present Value of a Series of Payments—Real Estate Capital Investment—Nominal and Effective Interest Rates—Internal Rate of Return—Future Value of Regular Deposits—Regular Deposits for Future Value—Depreciation Amount: Rate; Salvage Value; Schedule—Bond Present Value—Bond Yield to Maturity—Sale—Cost—Margin—Day of the Week—Moving Ad.
Education: Multiplication & Division—Add—Subtract—Fraction & Decimal—States & Capitals—States & Order of Entry—States & Abbreviation—Inventors & Inventions—World Capitals & Countries—Urban Areas & Population—Authors & Books—Presidents & Order—States & Largest City—Basenum.
Graphics: Left Right—Random Ad—Graphic—Blocks—Fireside—Snow—Step Ad—Step Ad 2—Launch—Ratrace—War Game—Weird—Herring—Blinker—Snoopy.
Home: Message Board—Expense Account—Nutrition—Mileage—Remember—Phone Codes—Night Check Off—Drunkometer—Perpetual Calendar—Babysitter—Calculator—Bartender—Christmas List—Vacation Check Off—Conversion.
Games: Speedy—Odd One—R. Roulette—Star Blazer—Search—Spyship—Tiger Shark—Jumble 2—Sting Ray—Stars—Sketch—Flipper—Scissors—Horse—Doomsday—Craps—Jumble 1—Mem. Quiz Letters—Mem. Quiz Numbers—Wheel of Fortune—Decision—Unjumble—Fifteen—Towers—Life—Star Trek—Race Track—Count—Roachrace—Gypsy.

The Library 100 is the best buy in software history! We do have one problem, though. Some people cannot believe that we can give you so much for only $49.50. We suggest that if you don't believe it, you ask your friends who have purchased this software package or go to your local computer store and ask to see it.

The Library 100 is a collection of 100 original, quality programs, plus a new, easy programming language for your TRS-80, Tiny PILOT (TPILOT).

It took us 37 pages in our manual just to say a little about each program, so we cannot describe them all here. We did list them. How many can you use?

You've read the announcements in Kilobaud and Creative Computing. It is here and it is real. A value in excess of $500.00 for only $49.50 plus $2.00 postage and handling (plus 4% for Georgia residents).

Watch for future software announcements. Hardware? Calculator? Business systems? Write and tell us what you want next.

The Bottom Shelf - Library 100

NOW AVAILABLE FROM 80-US

Save $2 postage & handling - only $49.50 postpaid

P.O. Box 7112, Tacoma, WA 98407 *Wash. Residents add 5.1% tax*

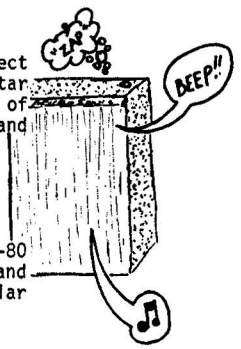
MasterCharge & VISA accepted - give full information -

realsoft™ COMPUTER CABLEVISION inc.
(202) 337-4691

air flight
from instant software
transform your trs-80 into a cockpit full of elaborate controls. take off, land, and do acrobatics once you learn how to fly.
4k level 1, 16k level 2 $ 7.95

golf
from instant software
18 holes of graphic golf with seven clubs. also classic peg game.
4k level 1, 16k level 2 $ 7.95

star warp
by mike partain
one of the first sound effect games. program based on 'star wars' with manual positioning of cross-hairs, game timer, and scoring.
4k level 2 $ 6.95



music
from software associates
enter scores to have your TRS-80 play music. better sound and easier to use than other similar program.
16k level 2 $ 9.95

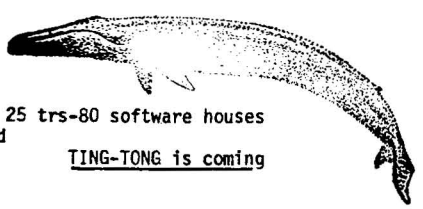
barricade
from small systess software
break through five-wall barricade in this machine language action game with 96 options.
4k level 1 or 2 $ 14.95

star fleet orion
from automated simulations
two-player tactical space game. create your own scenarios or use one of twelve described in battle manual. for serious gaming.
16k level 2 $ 16.95

beat the house
from 62
big-time casino games of chance: slot machine, craps, blackjack and roulette. results similar to las vegas.
16k level 2 $ 14.95

clinic
from 62
three programs for insights into your body, lifestyle and heredity: biorhythm, diet and longevity.
16k level 2 $ 14.95

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! !!
!! insiders™ tm !!
!! TRS-80 hardware journal !!
!! with machine software !!
!! $7.50 for six issues !!
!! !!
!! ask for sample with order !!
!! !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!



dealers for over 25 trs-80 software houses programs welcomed

TING-TONG is coming

DISASSEMBLER 1.1
from misosys
produces symbolic lables; output to video, tape or printer. editor assembler can load tapes. if you own editor/assembler, complete the package with this program.
16k level 2 $ 19.95

LEVEL 3
from G2 by microsoft
authors of level 2 basic take the next step adding more power to the trs-80. more graphic and editing commands; disk basic instructions without disk drives, and much more. place your order now.
16k level 2 $ 49.95

SYSTEM UTILITY
from software associates
two 'system saver' programs allow making backup copies of system tapes and loading any, even non-contiguous, system programs to disk. includes how to put microchess 1.5 on disk.
16k level 2 $ 14.95

G.S.F.
from racet computers
collection of fast, easy-to-use machine language routines. in memory sort with multiple variables and keys. array read/write to tape; compress, uncompress and move data. screen scrolling, save screen displays, plus. specify memory size in ordering.
level 2 $ 24.95

RENUMBER
from software associates
renumber programs! merge programs! all references are adjusted. for quality programming this machine language utility is a necessity. 16k, 32k and 48k versions for both level 2 and dos all on one tape.
$14.95

Computer Cablevision, inc.
2617 42nd st nw Washington DC 20007

☐ Yes, please send me:

title	price

master charge VISA postage: 1.00
total:

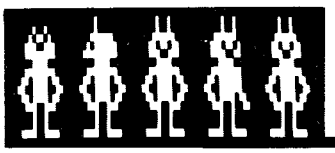
name: _____

address: _____
city, state
& code _____

☐ Check payable to
computer cablevision, inc.
☐ MASTERCHARGE mc bank code:
☐ VISA exp date:
card number:

signature:

CHINESE, ANDROID and OTHER NIMS



By Dr. A.R. Manson, North Carolina State University

Those readers captivated by the game "Android Nim" might be interested in knowing that it is a special case of the ancient Chinese game Nim. The most general game of Nim may have any number of rows and any number of elements (Androids?) per row. Determination of the winning player in the game depends only on two items, *after* one learns the winning strategy. These two items are: (1) which player moves first, and (2) whether a player "win" is defined to occur when the player removes the last element or when the player forces the opposing player to remove the last element.

The general rules are simple, in that a player may remove any number of elements from any single row in his turn, so long as he removes at least one and at most the number remaining on the row selected.

Let us assume that a win is defined to occur as in the "Android Nim" version, i.e., the player wins when he is able to remove the last element in his turn. Then the general solution to the r row problem with C_i elements on the i_{th} row is as follows. Express the original number of elements (the C_1) on each row as a binary number (0's and 1's).

If the number of 1's in *each* column of the r rows of binary numbers is even, then the player who goes first will lose, provided his opposing player uses the following simple strategy: Whatever number of elements are removed by the first player, the second player should re-express the remaining C_1 as rows of binary numbers, scan the columns of these binary numbers, and remove sufficient elements from that row which will make the binary representations of the C_1 have an even number of 1's in all columns. Continuation of this strategy will always result in a win for the player who goes second.

If the number of 1's in at least one column is odd, then the player who takes the first turn can always win. This is accomplished by removing the appropriate number of elements from that row which will make the binary representations of the C_1 have an even number of 1's in all columns. After the opposing player removes elements, the aforementioned strategy is repeated until the win is obtained.

This process is easy to demonstrate. For example. Android Nim has r = 3 rows with $C_1 = 7, C_2 =$, and $C_3 = 3$. In terms of binary numbers these C_1 may be expressed as:

1 1 1 = 7
1 0 1 = 5
0 1 1 = 3

Note that the third column of the binary representations of the C_1 contains an odd number of 1's (i.e. 3). Thus the first player merely removes one element from any of the three rows to give an even number of 1's in each column. This leads to:

1 1 0 = 6 1 1 1 = 7 1 1 1 = 7
1 0 1 = 5 or 1 0 0 = 4 or 1 0 1 = 5
0 1 1 = 3 0 1 1 = 3 0 1 0 = 2

Where all of these possible situations have an even number of 1's in each column. Player number 1, thereafter can force a similar situation regardless of player 2's choice and therefore can always win. Try it!

Now let us expand the game to one having r = 4 rows with $C_1 = 7, C_2 = 5$, $C_3 = 3$, and $C_4 = 1$ elements per row. The binary representation is:

1 1 1 = 7
1 0 1 = 5
0 1 1 = 3
0 0 1 = 1

which has an even number of 1's in all columns. Thus the *second* player to select can always win, regardless of what selection the first player makes, by following the aforementioned strategy.

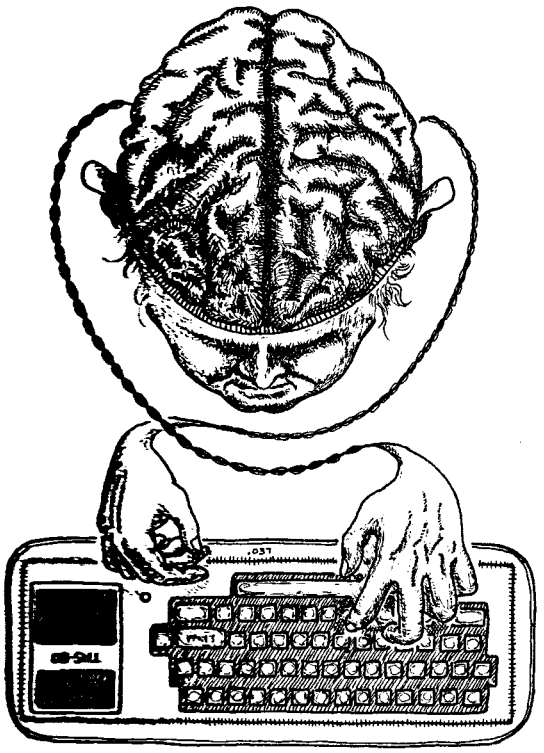
In the two games illustrated, the number of elements per row (C_1) were chosen to be odd numbers. However, the player's optimum strategy works for even C_1 as well. An even number of 1's in *all* columns of the binary representations of the C_1 is a losing position for the player whose turn it is, while an odd number of 1's in at least one row is a winning position for the player whose turn it is.

If a player win occurs when the opposing player is forced to remove the last element, as in the ancient game of Nim, then the game is merely a mirror image of the above version.

Whether these comments will lead to programmers adding more (and necessarily smaller) Androids per row and more rows either in a deterministic fashion or in a random fashion will be interesting to see. It would be a simple matter to generate a game having a random number of rows from say 3 up to 10 and to generate a random number of elements (of some kind) per row from say 1 to 20. it is hoped that the game solution given here will not detract from the player's enjoyment of Nim; however, there is no way to make Nim a fair game by changing r or the C_1, except by alternating which player has the option of the first move.

There are many similar mathematical "tricks" which can be used as the basis of numerical games. For example, the reader may want to attempt programming a game having a set of 50 elements (graphical figures). Players alternate in turn as in Nim and may remove any of the 50 elements in their turn, so long as they remove from one to six elements. The player forced to remove the last element loses the game. Thus, the game is similar to Nim except that there is no row restrictions. The player who goes second in this game can *always* win, and the winning strategy in this game is even simpler than that of Nim. It's solution is left as an exercise for the minds of the curious.

WHICH BRAIN?



Our feature article for this issue was a concerted effort in the exploration of an exciting application for the Microcomputer. Consulting Services Northwest did the actual program, which was extensively tested and validated by Dr. Livingston. Credit for the cover drawing goes to Leo Christopherson. Dr. Livingston, we presume, is already working on three other tests, which when completed, should give a complete personality profile. The program with this article requires 16K , Level II; printer is optional.

By Dr. G. B. Livingston, Tacoma, WA

The operation of a computer has often been compared to the operation of the human brain. Attempts have even been made to simulate the operation of the brain with computers. The computer is a relatively simple system, however, by comparison with the brain. In fact, we actually have TWO brains; a LEFT brain which controls our intellectual processes, and a RIGHT brain which controls our intuitive processes.

The left brain is the home of the intellect. It deals with facts and logic. It makes deductions, categorizes and sorts data, and reasons out solutions to problems.

The right brain houses intuition. It is the center of imaginative and creative thinking and the area of possibilities rather than realities. It is where we develop ability to sense where to put our hand while catching a fly ball when we cannot compute its final position.

Everyone uses both sides of the brain. Very early in life we develop a preference for left or right brain processing. This preference becomes the dominant function. The other side assumes an auxiliary role.

The preference that we develop for left or right brain processing will strongly affect our lives. How we solve problems and resolve crises will depend on our preference. Our choice of job and the relative success that we enjoy is strongly affected by our brain pattern. Our personality patterns and "style" and the nature of our relationships with other people will be a direct result of our pattern.

There are distinct differences between the left and right brain dominant thinking processes. Left brain processes operate at the level of "conscious awareness," but intuitive processes operate below this level with only the final result surfacing to consciousness. The left brain is slow, often imprecise. Each operation is under conscious control. Right brain perception is quick, reaching conclusions with no visible effort and no apparent connection.

The left and right hemispheres of the brain are connected by a few hundred million nerve fibers which serve to coordinate the operation of the two brains. Language resides in the left brain and the right must borrow to do its processing. Many psychologists also believe the right brain borrows from the left's analytical ability to do its processing below the conscious level.

Differentiating between two basic brain patterns does not imply that one is better than the other. As with most things in life, each pattern of thought has its own strengths and weaknesses.

Psychologists have studied left and right brain functioning for years to discover the clues that can help to identify the dominant processing method. There are two tests that have been developed to determine the type of perception which is dominant. These are the Myers-Briggs indicator and the Gray-Wheelwright indicator. Both are paper and pencil tests and have been well tested. The program which follows is based on the "Semantic Differential" test first developed by Charles Osgood.

The test in the program determines your thinking pattern by presenting you with a choice between two opposing adjectives which have been related to intuitive and analytical thinking processes. Placing the marker between the adjectives to balance their relative appeal to you establishes a "score" which the program then uses to determine your thinking pattern. The adjectives are like unequal weights on a balance, with the marker as the pivot point. When the marker is placed so that the balance is perfect, the result is recorded. The program has been validated against the Myers-Briggs test and should provide a good evaluation of

MIND READER

Here is a clever little treat for Level I. It will read your mind, and you do not have to input any other numbers except the seed number in line 3. You just push enter, and keep track of your numbers mentally. The computer will tell you the answer! It has nice visual effect and care should be taken to type in lines 110 and 112, using the exact spacing shown in the listing. Now go ahead and type in the program, and meet Mervin the Mind Reader.

By Jerry Atkinson, Snowville, UT

```
1 CLS
2 REM "MIND READER" BY JERRY ATKINSON, SNOWVILLE UT.
3 IN."ENTER A NUMBER BETWEEN 1 AND 100";N
4 IFN>100IN."ENTER A NUMBER BETWEEN 1 AND 100, DUMMY!";N:G.4
5 F.J=1TON:X=RND(32000):N.J
6 CLS
10 IN."HI.  I'M MERVIN.  WHAT'S YOUR NAME ";A$:P.
12 P."HELLO ";A$;".  I'M GOING TO READ YOUR MIND."
14 P."TO DO THIS, I WILL ASK YOU TO THINK OF A NUMBER"
16 P."AND THEN DO SOME SIMPLE ARITHMETIC WITH IT."
18 P."THEN I WILL TELL YOU WHAT YOUR ANSWER IS"
20 P."WITHOUT KNOWING WHAT NUMBER YOU STARTED WITH.":P.
22 P."IF YOU WANT TO USE A CALCULATOR, THINK OF ANY NUMBER"
24 P."WITH 6 DIGITS OR LESS.  IF NOT, MAKE IT SIMPLE--"
26 P."SAY LESS THAN 50 OR SO.":P.:G.28
27 P."OK ";A$;", LET'S DO IT AGAIN."
28 P."IF YOU USE A CALCULATOR, ENTER A  '' 1 '' ,"
30 IN."OTHERWISE, ENTER A  '' 2 '' . ";Q:CLS
32 IFQ=1 .50
34 IFQ=2T.38
36 G.22
38 X=RND(30):X=X*2:G.60
50 X=RND(32000):X=X*2:IFX<1000G.50
60 P.:P."REMEMBER, I WILL ONLY TELL YOU WHAT YOUR ANSWER IS,"
62 P."NOT THE NUMBER YOU THOUGHT OF."
64 P."DON'T TYPE IN ANY OF YOUR NUMBERS--- THAT MIGHT GIVE"
65 P."ME A CLUE!":P.
66 P."NOW THINK OF YOUR NUMBER.  WHEN YOUR READY, HIT ENTER"
68 IN."AFTER EACH QUESTION MARK. ";B$:CLS
70 IN."NOW DOUBLE YOUR NUMBER.    ";B$:P.
80 P."           ADD  ";X;:IN.B$
90 P.:IN."OK--  NOW DIVIDE BY 2  ";B$:P.
100 IN."AND SUBTRACT THE NUMBER YOU FIRST STARTED WITH.   ";B$
102 F.I=1TO101
110 P.AT579;"O****-*****                                    VIBRATIONS"
112 P.AT579;" OOOOO-OOOOO  I'M BEGINNING TO PICK UP YOUR  VIBRATIONS"
116 N.I
120 F.N=1TO900:N.N:CLS
140 P.AT399;"I'VE GOT IT!  YOUR ANSWER IS"
142 F.D=1TO800:N.D
145 F.H=1TO6
146 P.AT538;"       "
147 F.G=1TO60:N.G
```

(Continued on page 25)

A NOTE ON BASIC

Have you ever tried to use the INKEY$ function to test for the ENTER key or one of the arrows? We don't mean the standard null test everyone uses,

IF INKEY$ <> "" THEN

Instead, we are referring to a real test that might branch one way on recognizing an arrow, and another on seeing ENTER. Trying to do this with a string comparison doesn't work (what character is an ENTER?). The CHR$ function doesn't work either, but a comparison to the numeric ASCII value given on page C1 of the Level II reference manual will work. For example, the ENTER key has an ASCII number code of 13, so the following routine will test for the ENTER key:

```
10 C$ = INKEY$:IFC$ "" THEN 20ELSE10
20 C = ASC(C$)
30 IFC = 13THEN40ELSE10
40 PRINT"YOU PRESSED THE ENTER KEY"
50 END
```

Study appendix C in the manual to find the codes for other keys. A good use for graphics displays is being able to use the arrows to position a marker on the screen. To do this, you can test ASC(C$) for the value of 24 (left arow) 25 (right arrow), 26 (down arrow), or 27 (up arrow). The test can branch to routines that will move the marker in the correct direction.

It's important to note the use of the intermediate string variable C$. Without C$, you would be testing only what you held down while you held it. This can cause disruption in a game since what you expect as response may not happen if you remove your finger too soon. Even more of a problem occurs if you have to test for multiple branches, because then the time factor is important enough, that the program might check for two branches, but not the third because the key is no longer pressed.

```
148 P.AT538;X/2
149 F.F=1TO200:N.F:N.H
150 Y=1:N=0:P.:P.
170 P.T.(6);"HOW 'BOUT THAT!  ";
172 P."DO YOU WANT TO TRY AGAIN ";A$;"?  Y/N ":IN.C
190 IFC=0T.215
200 CLS:IFC=1T.27
210 G.150
215 CLS:P."PRETTY NEAT HUH?  SEE YA LATER, ";A$;"."
230 END
```

VIEW FROM THE TOP OF THE STACK

By T. Rosenbaum, Technical Editor

The stack is the most important aspect of machine language programming to understand, since it is used to govern the mechanics of subroutine usage. A subroutine is a sub-program used to perform a function which is likely to be required by the main program several times. Examples of functions which would be good subroutines are:

A) A sub-program which would multiply the contents of the H register pair by the contents of the L register and return the results in the BC register pair.
B) A sub-program which would take the BCD (Binary Coded Decimal) number in the lower four bits of the accumulator and return its ASCII equivalent in the B register.
C) A sub-program which will take an ASCII character in the A register and return its equivalent IBM Selectric Correspondence Code character in the A register.

Subroutines are very important, since they save many bytes of memory. They simplify programming considerably. Imagine how many bytes would be wasted in a mathematical package (group of programs) if you had to load the machine language code for a multiplication into the machine each time a multiplication was called for.

Last issue, we learned how to use the JUMP commands so we could build a multiply program and JUMP to it. The problem is how to get back to the main program once you have JUMPed to the sub-program. In order to get back to the main program (RETURN) after you have jumped to the sub-program (CALL), the Z80 must be able to remember where it was when you CALLed the sub-program. The Z80 is able to remember where to RETURN to after subroutine execution by storing the RETURN addresses in the STACK.

The stack may be located anywhere in RAM (it is simply a designated area of RAM). It must be located in RAM, never in an area occupied by ROM. Level I, II and DOS BASIC always locate the stack at the high end of available memory, i.e., at 7FFF for a 16K machine, BFFF for a 32K machine and FFFF for a 48K machine.

The location of the stack is defined by using a special 16 bit register called the STACK POINTER(SP). It contains the address of the top of the stack (and is where we got the name for this column).

Data is moved to and from the stack in a LIFO (Last In First Out) manner. This means that the last data written into the stack is the first data removed when data is read from the stack. Let us try an example: Assume the current address of the stack pointer is 8000 and we load 23C7 into the stack (the stack stores absolute addresses, so whenever you store or recall data from the stack, the data is transferred in 2 byte chunks) after this operation is performed the stack will look like:

Address	Data
7FFF	C7
7FFE	23 ← SP

and the SP will contain the address 7FFE. When data is stored into the stack, the operation is called PUSHing the data onto the stack. When data is retrieved from the stack it is called POPing it from the stack. Let us continue the above example by PUSHing 457E and 8014 onto the stack.

7FFF	C7
7FFE	23
7FFD	7E
7FFC	45
7FFB	14
7FFA	80 ← SP

and the SP will contain 7FFA. If one address is POPed off the stack, the stack will look like:

7FFF	C7
7FFEE	23
7FFD	7E
7FFC	45 ← SP
7FFB	14
7FFA	80

and the SP will contain 7FFC. Notice that the data POPed off the stack from locations 7FFA, 7FFB has been left unchanged.

Note that the low order byte of an address is PUSHed onto the stack first followed by the high order byte second.

The Z80 uses the stack to govern the orderly flow of program control during execution of subroutines. Subroutines are implemented by using the statements CALL nn and RETURN. The CALL nn statement causes three operations to occur in the Z80:
1) The contents of the PC are incremented by two and PUSHed onto the stack.
2) nn is stored in the program counter (PC).
3) The SP is decremented by two.

The RETURN statement causes two operations to take place:

1) The top address on the stack is POPed off and loaded into the PC.
2) The SP is incremented by two.

After a CALL nn instruction is executed, the next instruction executed will be the one located at address nn, for example consider the following section of memory, (XX = value of the byte is unimportant, or don't care):

B012	CD
B013	00
B014	52
B015	XX
B016	XX
B017	XX
B018	XX

SP = B018
PC = B012

Following the execution of the next instruction in this sequence (CALL 5200), the above section of memory will be changed to:

B012	CD
B013	00
B014	52
B015	XX
B016	18
B017	B0
B018	XX

SP = B016
PC = 5200

and the next instruction to be executed would be whatever instruction was stored at 5200. If the instruction stored at 5200 was CALL 60B2 and it was executed, the above section of memory would read:

B012	CD
B013	00
B014	03
B015	52
B016	18
B017	B0
B018	XX

SP = B014
PC = 60B2

Note the instruction which was located at B012 (CALL 5200) has been destroyed. When the address 5203 was PUSHed

onto the stack by the CALL 60B2 instruction, it replaced the 52 of CALL 5200 with 03. Now the instruction at B012 is CALL 0300. Obviously, this will cause your program to bomb if control returns to B012 expecting to find a CALL 5200.

The above example is a good demonstration of one of the largest danges involved in using the stack. The Z80 cannot tell if the memory in which the stack is located contains a program or is free memory. Accordingly, steps must be taken to prevent the stack from "eating" into your program. There is no "iron clad" method of protecting your program from the stack and allowing an unlimited number of nested subroutines (a subroutine is "nested" if it is called by another subroutine) since the Z80 cannot address an infinite amount of memory. Happily, even the most elegant of programs (such as level II BASIC) will only have subroutines nested about 100 levels deep at most. When we say subroutines are nested 100 levels deep, it means that 100 CALL statements are executed before a RETURN statement is executed.

The method most often used to protect programs from the stack is to form a buffer in memory under the stack, which will allow the stack space in which to grow. Each level of nesting will increase the size of the buffer by two bytes. Be conservative when you choose the size of the buffer in order to allow for future program expansion or unexpected branches in your program. Unexptected branches in your program may occur if you CALL subroutines located in the Level II ROM without knowing how the subroutine works - it may PUSH many addresses onto the stack and POP them off before returning. The SP will be in the same place it was before the subroutine was called, but the stack may still have "eaten" your program if you didn't provide a large enough buffer.

It is just as important to protect the stack from your program as it is to protect your program from the stack. Once you set up your buffer area for the stack, you must not use the buffer to store data. If you do, the RETURN addresses which have been stored in the stack by CALL statements will be destroyed and the program will fail.

There is an instruction which is a specialized CALL instruction called the RESTART (RST) instruction. It is a one byte instruction of the form RST p, where p is an integer from 0 to 7. The RST instruction forces a CALL to one of eight addresses in RAM. the RST's and their respective CALL addresses are as follows:

RST 0	0000H
RST 1	0008H
RST 2	0010H
RST 3	0018H
RST 4	0020H
RST 5	0028H
RST 6	0030H
RST 7	0038H

The purpose of the RST is to save memory space. A normal CALL requires three bytes each time it is used. With an RST you can put a JUMP at the RST address and create a one byte CALL to any address in memory. The RST is used to CALL frequently used subroutines. Level II BASIC uses RST1 for checking syntax, which is done many times in Level II Basic. Each time an RST instruction is used, two bytes of memory are saved which is a big savings of memory in a complex program.

There are several instructions which will affect the stack by either transferring data to or from it or changing the Stack Pointer. These instructions and a brief explanation of how they function are listed below.

1) CALL nn	A Call to address nn, as explained earlier.
2) CALL nn,cc	A Call to address nn if the condition cc is true. Otherwise the CALL is not executed. cc may be Z, NZ, C, NC, PE, PO, P, or M which are states of the flags as explained in the last issue.
3) RET	Unconditional RETURN.
4) RET cc	Conditional RETURN. A Return will be executed if cc is true.
5) RST p	Restart instruction, explained earlier.
6) EX (SP), HL	The L register is exchanged with the memory location whose address is contained in the SP and the H register is exchanged with the memory location whose address is (SP + 1)
7) EX (SP), IX	Similar to 6, except the exchange is with IX instead of HL.
8) EX (SP), IY	Similar to 6, except the exchange is with IY instead of HL.
9) LD SP,dd	The contents of register dd are loaded into the stack pointer. dd may be HL, IX or IY
10) LD SP,nn	The data nn are loaded into SP.
11) LD SP, (nn)	The data pointed to by (nn) are loaded into SP
12) PUSH dd	The data in register dd are PUSHed onto the stack. dd may be AF, BC, DE, HL, IX or IY.
13) POP dd	The top addresses from the stack are POPed off and loaded into register dd.

We will continue with the STACK in the next issue. Also, we will then review the projects assigned in the last issue and assign new projects involving the stack.

"Gee Whiz"

By T.R. Dettman, Associate Editor

Here is another of those clever ideas which makes you say, "Wow, why didn't I think of that?" Keep in mind that the program which this program calls must be modified to check locations 32767 for a 1, and if it finds it to give the command to RUN "GEEWHIZ".

Funny how some of the ideas developed for games, find application in business. This idea can be put to very good use in a business application, especially where memory cannot hold the entire program. Of course, you would want to dispense with the commentary, and possibly use LOAD "Main program", R if you want to maintain open files between the various sub-programs.

Everyone who sees my computer is anxious to try it for themselves. Rather than give them a short course in computer programming, or waste time trying to load a bunch of programs for them, I have a little program which has a conversation with them and then gives them options for games or displays.

I call it "GEEWHIZ" (in fact, that is it's filename). There is nothing particularly complex about the program, but there are some neat features in it.

I wanted the program to function as a master controller which all other programs would return to if called by it. This way, my guests would choose to play more games if they wished, and all of the program calls would be done by the master program.

To do this I use location 32767 (16K machine) and POKE 1 into that location at the beginning of the program in statement 40. Each game on the disk which is configured to operate with the GEEWHIZ program has been modified so that before it comes to an END statement, it PEEK's at this location and if it finds a 1 there it reloads the GEEWHIZ program and executes it. (Example: A=PEEK(32767): IF A=1 THEN RUN "GEEWHIZ")

The GEEWHIZ program in it's turn checks this location each time it is run (in statement 30) and if it finds a 1 there, it knows that it is in the second or later cycle of it's run. It then skips over the conversation part and goes right to the list of available game programs. In order to do this, you must answer MEMORY SIZE? as some number less than 32767 (I usually use 32760). This will keep that location free and available for the "pass down" bit.

Statements 50 to 80 use the graphics capability on the screen to display HI in large capital letters. After that, the conversation starts with a very few questions just to get a little warm-up. After that it goes directly into the game menu.

When the program is terminated normally by selecting option 100 in statement 430, the program POKE's a zero back into location 32767 so that any other games that are run will not automatically transfer back into GEEWHIZ.

The program is trivial for the average programmer, but a source of unending wonder to the uninitiated, who don't think they could ever understand how to operate a computer●

```
10 REM GEE WHIZ INTRODUCTORY PROGRAM 1.0
20 REM (C)1979 80-NW PUBLISHING CO *
30 IF PEEK(32767)=1 THEN 340
40 POKE32767,1
50 CLS:FORY=12TO26:SET(34,Y):SET(35,Y):SET(36,Y)
60 SET(50,Y):SET(51,Y):SET(52,Y):NEXTY
70 FORX=36TO49:SET(X,18):SET(X,19):SET(X,20):NEXTX
80 FORY=12TO26:FORX=65TO67:SET(X,Y):NEXTX:NEXTY
90 PRINT@720,"I'M TERRY'S COMPUTER"
100 FORI=1TO2000:NEXTI
110 CLS:INPUT"WHAT'S YOUR NAME";A$
120 PRINT:PRINT
130 PRINT"I'M PLEASED TO MEET YOU ";A$
140 INPUT"WHERE ARE YOU FROM";B$
150 PRINT:PRINT
160 PRINT"I ONCE KNEW SOMEONE FROM ";B$;" !"
170 PRINT"DO YOU KNOW FRITZ THE"
180 PRINT"PLUMBER? HE'S FROM ";B$;" TOO!"
190 FORI=1TO1000:NEXTI
200 PRINT:PRINT:PRINT:PRINT
210 PRINT"GEE ITS GREAT TO FIND SOMEONE FROM ";B$
220 PRINT"WHO I CAN REALLY TALK TO ABOUT OLD TIMES."
230 FORI=1TO2000:NEXTI:PRINT:PRINT
240 PRINT"YOU KNOW ";A$;", IT REALLY GETS TIRING TALKING"
250 PRINT"ONLY TO TERRY. HE'S SUCH A BORE. YOU KNOW THE TYPE,"
260 PRINT"ALL WORK AND NO PLAY, AND THE MATH, JEEEEZE --"
270 PRINT"I DON'T UNDERSTAND EVEN HALF OF IT."
```

```
280 FORI=1TO2500:NEXTI
290 CLS:PRINT"HOW ABOUT SOME GAMES? I'LL LET YOU LOOK AT"
300 PRINT"A LIST OF SOME OF MY FAVORITE GAMES AND WE CAN"
310 PRINT"PLAY A FEW. PICK THE NUMBER BELOW FOR THE GAME YOU"
320 PRINT"WANT TO PLAY AND I'LL DISH IT UP FOR YOU"
330 PRINT:PRINT:FORI=1TO2500:NEXTI
340 CLS:PRINT"CURRENT GAMES"
350 PRINT"1","PROG#1":REM * PUT YOUR PROGRAM NAMES HERE AND
360 PRINT"2","PROG#2":REM * IN THE DATA STATEMENT IN
370 PRINT"3","PROG#3":REM * LINE 530
380 PRINT"4","PROG#4"
390 PRINT"5","PROG#5"
400 PRINT"6","PROG#6"
410 PRINT:PRINT"INPUT 99 TO GET A CONTINUATION OF THE LIST"
420 PRINT"INPUT 100 TO END THE PROGRAM"
430 INPUT"MY CHOICE IS";G
440 IFG=99THEN470
450 IFG=100THEN540
460 GOTO500
470 CLS:PRINT:PRINT"SORRY, NO MORE PROGRAMS JUST NOW"
480 PRINT"I'LL GO BACK TO THE ORIGINAL LIST SO YOU CAN"
490 PRINT"SELECT SOMETHING":FORI=1TO2000:NEXTI:CLS:GOTO340
500 FOR I=1TOG:READ Z$:NEXT I
510 RUN Z$
520 END
530 DATA "PROG#1","PROG#2","PROG#3","PROG#4","PROG#5","PROG#6"
540 POKE32767,0
550 END
```

SYSTEM/COMMAND

By Phil Pilgrim

By now it is no secret that the TRS-80 can produce sound without hardware modifications. Here is an assembly program for getting that sound from Level II BASIC. It is called via USR(0) and requires a character string holding a sequence of tone data as input. The audio signal is available on the cassette plug normally inserted into AUX and may be amplified with any audio amplifier.

The format of the tone sequence is shown in Fig. 1. Each tone is defined by a duration character followed by a pitch character. The longest duration is CHR$(255) (about 1.2 seconds); the shortest, CHR$(1). The highest pitch is CHR$(1); the lowest, CHR$(255). By concatenating these duration/pitch pairs into a string, a tone sequence-- or melody--can be obtained.

In order for SOUND (Fig. 2) to reproduce a tone sequence, one has to tell it where that sequence is in memory. This is done by selecting a string name, finding its "dope" vector using VARPTR, and storing the location of this vector in the area called SBUFF. Thereafter, SOUND finds the string characters it needs to produce the tones, keeping the next character location in HL, and the number of characters remaining pushed onto the stack. Fig. 3 illustrates the sequence of address pointers involved.

The tones themselves are produced by alternately, outputting 1's and 3's to the cassette port (OFFH). The period between these outputs is determined by the pitch character (kept in the C register). The overall length of the resulting tone comes from the duration character, which is counted down in register DE. The outcome is a square wave of the desired pitch and duration. (See Fig. 4).

To use SOUND, assemble it as shown using the Editor/Assembler. Dump it to a SYSTEM tape and reinitialize BASIC. for MEMORY SIZE? enter 32700. Then reload the SYSTEM tape, and type /Enter to set up the USR(0) pointer and get back to BASIC. The sample BASIC program (Fig. 5) illustrates how to handle the VARPTR results. It also plays a little tune. It does this by converting the symbolic duration and pitch strings, D$ and P$, into a tune sequence (S$) using the even-tempered scale contained in array TN. You can experiment with different melodies by modifying D$ and P$. Happy listening!

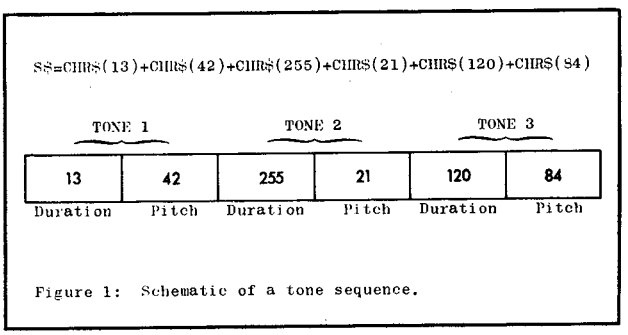


Figure 1: Schematic of a tone sequence.

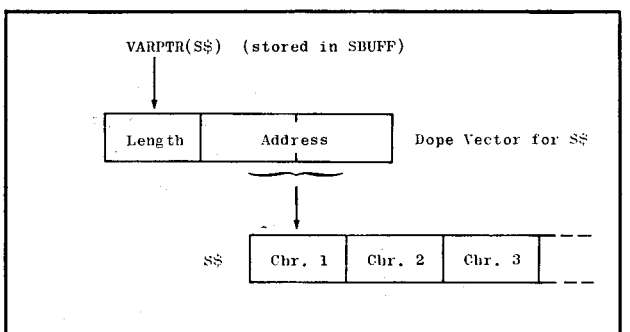
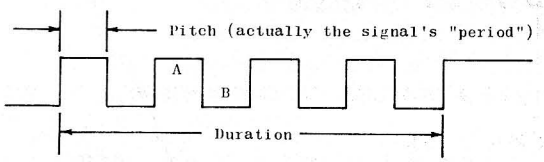


Figure 3: Addressing arrangement for strings.

```
        ORG     7FC4H           ;NEAR END OF 16K MEM.
START   LD      HL,SOUND        ;SETUP USR(Ø) TO
        LD      (16526),HL      ;POINT TO SOUND
        JP      1A19H           ;JUMP TO BASIC
SOUND   LD      HL,(SBUFF)      ;GET STRING DOPE VCTR
        LD      A,(HL)          ;GET LEN
        OR      A               ;IS IT ZERO?
        PUSH    AF              ;SAVE LEN AND TST RSLT
        INC     HL              ;NEXT ITEM
        LD      E,(HL)          ;LSB OF STR ADDR.
        INC     HL              ;NEXT ITEM
        LD      D,(HL)          ;MSB OF STR ADDR.
        EX      DE,HL           ;STR ADDR IN HL
NEXT    POP     AF              ;NO. CHRS REMAINING
        RET     Z               ;NO MORE
        DEC     A               ;MUST BE 2 OR MORE
        RET     Z               ;IT'S NOT
        DEC     A               ;COUNT 2ND IN PAIR
        PUSH    AF              ;SAVE REMAINING CNT.
        LD      D,(HL)          ;GET DURATION CHR.
        LD      E,Ø             ;(*256)
        INC     HL              ;POINT TO PITCH CHR.
        LD      C,(HL)          ;GET IT
        INC     HL              ;NEXT CHR, IF ANY
TONE    LD      A,1             ;SQUARE WAVE PLUS
        CALL    HOLD            ;SUSTAIN IT
        JR      Z,NEXT          ;ALL DONE
        LD      A,3             ;SQUARE WAVE MINUS
        CALL    HOLD            ;SUSTAIN IT
        JR      NZ,TONE         ;MORE OSCILLATIONS
        JR      NEXT            ;ALL DONE
HOLD    OUT     (ØFFH),A        ;TO CASSETTE PORT
        LD      B,C             ;PITCH PERIOD
LOOPB   DEC     DE              ;DECREMENT DURATION
        LD      A,D             ;TEST FOR ZERO
        OR      E               ;       "
        RET     Z               ;DONE IF SO
        DJNZ    LOOPB           ;CNT DOWN PERIOD & LOOP
        RET                     ;DONE COUNTING
SBUFF   DEFS    2               ;THIS IS LOC. 32766
        END     START           ;FOR AUTOSTART
```

Figure 2: The tone-generating program SOUND.



Outputs to cassette port (ØFFH): A=1, B=3.

Figure 4: Diagram of cassette AUX output showing Duration and Pitch.

```
1Ø CLEAR 1ØØØ:DIM TN(26):S$="":V=VARPTR(S$)
2Ø V1=INT(V/256):VØ=V-V1*256:REM BREAK V INTO BYTE-SIZED
   COMPONENTS.
3Ø POKE 32766,VØ:POKE 32767,V1:REM POKE V INTO SUFF.
4Ø FOR T=1 TO 26:REM CREATE EVEN-TEMPERED SCALE.
5Ø P=.ØØ4*1.Ø5946↑T:P=INT(1/F):TN(T)=P:NEXT
6Ø D$="333311121633331112161111211360212114121212120Ø111216"
7Ø P$="YVRMOQRORMTYVROQRTVTVWVTYVTRRTVROPOMMRVTMRVTVWYVRTMR"
8Ø FOR I=1 TO LEN(D$):REM CREATE TONE SEQUENCE IN S$.
9Ø DU=(ASC(MID$(D$,I,1))-48)*2:IF DU=Ø THEN DU=1
1ØØ S$=S$+CHR$(DU*2Ø)+CHR$(TN(ASC(MID$(P$,I,1))-64))
11Ø NEXT:K=USR(Ø):REM PLAY THE TUNE IN S$.
```

Figure 5: A BASIC program which plays a tune using SOUND.

To the readers of the 80-U.S. Journal ...for your ideas, comments and support. Keep 'em coming, folks. We aim to please!

HANGUPS.....

Here is an odd one from Gary Bellow in Salem, Oregon who says: "I have found on more than 3 disk based TRS-80 systems that whenever a basic program is saved on disk - regardless of the memory Size or bytes cleared for string storage - that if the basic program is exactly 5885 bytes long it will give an INTERNAL ERROR when you try to load it back from the disk. Also, I have been unsuccessful at getting the APPEND utility to work".

We tried this and he is right. With a basic program of exactly 5885 bytes it does just that. Also, we noted after loading the program back in, that the program was in memory, intact, but when we try to remove one byte by editing, the disk kicks back on and locks the system up. The number 5885 is not, as far as we can see, an exact multiple of any significant number connected with the size of the sectors. Changing the size of the program just one byte makes all well again. It happens, we don't know why, either, but this may explain some of the strange things happening from time to time. Anyone have ideas?

As for APPEND: It does not work in DOS 2.1, nor did it work in 2.0, maybe it will be fixed in Version 2.2. Although APPEND shows on your 2.1 DOS when you ask for LIB, the new DOS manual from Radio Shack fails to even mention it.

.....

Here is one from Dr. Wilbur A. Muehlig, Omaha, Nebraska who says he has been modifying a text editing program in basic for the 80, including adapting it to disk and the substitution of the Line Input for Input. Since he had 32K available, he changed the Clear statement from 2000 to 10000. After the changes the program became intermittent when reviewing the text. It would completely lock up, not even responding to the reset button. It was found that theClear statement was causing the problem, that if he cleared 8000 instead of 10000 it worked. He says:

"Apparently with the program (about 13K) and arrays, I was running out of memory. Instead of telling me, the old TRS went into the hysterical cataleptic trance."

Dr. Muehlig's problem was apparently using so much space in memory that there was no room for the stack. Could be that without the stack to point the way it couldn't even direct Basic to the proper error message.

.....

From Gerald Clark, Newport, Maine comes yet another problem on Bowling (Jan-Feb 79 80-US). "We keep getting back an error on line 130 (SET X,Y) and can't find it. Perhaps you can give us a clue as to where it might be."

Make sure your data statements in lines 4990 through 5020 are correct. Also that you are using the zero a 0 not the "oh".

.....

If you have a TRS-80 Disk System, you have a good little machine language monitor. Even if you have never programmed in machine language before, this article will show you how easy it is.

The Monitor You Already Have

By George Blank, Leechburg, PA.

First, let us examine the monitor. It is really not called a monitor, but the DEBUG routine in the Disk Operating System. If your system is up and running, use the on/off switch to turn the computer off for ten seconds. That is the little black button on the back of the computer beside the power connector. When you turn it back on, the memory should be cleared and then the DOS will load immediately. As soon as the DOS is ready, enter DEBUG. Press the break key and the screen should fill with characters. These characters include a listing of each register and its contents in Hexadecimal Code, a listing of the stack (an area of memory set aside for saving information used by the registers), and a display of a block of memory.

At the upper left corner of your screen is the display for registers A and F. The condition codes are displayed beside them. The last register displayed is the Program Counter, identified as PC. The program counter tells the computer which instruction to execute next. After the registers is a display of 64 bytes of memory in rows of 16 bytes. The address of the first byte in each row is displayed on the left.

Let us take a look at some of the features of the DEBUG routine. First, press the A key. This converts the memory display from Hex to ASCII. Letters are usually stored in memory in ASCII codes, and so are the graphic characters. Now press the H key. This converts back to Hex code. Play with A and H a few times and look at particular characters to see them in both forms. If there is no ASCII character for a given Hex pair of numbers, the computer displays a period.

Each byte of memory can be one of several items. It can be an operation code, an address, or data. The operation codes are instructions telling the computer to do something. The addresses tell the computer where to find data, store data, or go to the next instruction. The data can be a variable, a letter to be displayed, a number, or a graphics code. As an example, the Hex code 41 could be an instruction telling the computer "take the number in register C and put it in register B". It could also be the letter A in ASCII. If it is an integer variable, it is the number 16384. If it is a Binary Coded Decimal (BCD) variable, it is, believe it or not, the number 41. If it is a Hex variable, it is the number 65. As an address, it could be combined with another byte to form one of the 64K bytes of memory that the Z80 chip can address, or it could be relative address telling the computer to skip the next 64 bytes of memory and go to the 65th. It could also be a number that is to be added to one of the index registers to form an address.

I hope that explanation did not hopelessly confuse you. Any one byte of memory can be many different things. The program tells the computer how to use each byte, but we can begin by saying that the computer always considers the first byte it comes to an operation code and unless an operation code tells the computer what to do with a byte of memory, it considers all data to be operation codes.

Back to the other features of our DEBUG routine. Enter D000. This tells the computer to display the first 64 bytes of memory at the bottom of the screen. You are now looking at the contents of the your Level II ROM. Look at it in ASCII and Hex with the keys A and H. Now that you have become expert with these two keys, let us learn a new one.

Press the letter S. This eliminates the display of the registers and the stack and uses the whole screen to display, not 64, but 256 bytes of memory. If you really miss the registers, you can get them back by pressing X.

The command X will get you back to the normal display at any time the full screen is displayed or in the middle of any incomplete instruction. For example, if you had started to type D000 and hit the wrong key so that you had D999, you could just enter X and start over. If you have sufficiently mastered X and S, you are now ready for two instructions that will work with either display.

Press ;. This advances the memory display and lets you see the next 64 or 256 bytes of memory. Now press -. This backs up the memory display 64 or 256 bytes, depending on the display mode.Obviously the ; was chosen because the same key has the + on it , so that you can remember "plus" or "minus" to get the memory display.

You are now entitled to a break. Using the commands you have already learned, play with your computer and look through the memory. When you get tired of that come back and we will learn some more. To help you, here is a summary of the commands we have learned so far.

"BREAK"	Begin DEBUG routine from DOS after entering DEBUG (to enter from BASIC, type CMD"D")
A	Display memory in ASCII code
H	Display memory in Hex code
Dnnnn	Display the memory beginning at address nnnn Hex
S	Use the full screen for display
X	Return to normal display or cancel incomplete instruction
;	Advance memory display one block
-	Back up memory display one block

Now that you have made good use of your break by comparing the display with the memory map in your Level II manual, looking at the ROM, the keyboard, the display, the DOS program, the vectors, the variables, and all that empty memory- now that you have figured out everything that goes on in your computer, let us do a little machine language programming. (That is what you did on your break, isn't it?)

To keep things simple, I would like you to learn 5 Hex Z80 operation codes. I will include the Assembly Language mnemonics and an explanation to make you an expert in both machine language and assembly language programming. Here is a table of your instructions:

00	NOP	Do nothing. This tells the computer to waste time.
AF	XOR A	Exclusive OR Register A. This tells the computer to compare all eight bits of register A with all eight bits of register A and put a 1 in any bit that is different. Just an easy way to clear register A.
3C	INC A	INCrement Reg A, add one.
3D	DEC A	DECrement Reg A, subtract one register A.
C3nnnn	JPnnnn	Jump to memory location nnnn. This loads nnnn in the program counter and causes the computer to execute the next instruction found at that address. The crazy thing about this is that we have to put the second byte first. IE, if we want address 5201 Hex, we will enter C3 01 52.

The program we will enter into the computer will clear register A, do nothing for one instruction, add one to register A, add one again, subtract one, then go back to the do nothing instruction and go through the cycle endlessly, except for clearing the register.

Choosing an arbitrary location in memory to begin, 5200 Hex, here is our program:

5200	AF	XOR A
5201	00	NOP
5202	3C	INC A
5203	3C	INC A
5204	3D	DEC A
5205	C3 01 52	JP 5201

Now let us enter it in our memory. If you still have the DEBUG routine on, enter X to get the normal display. We wnat to see what goes on in the registers. Enter D5200 to display the memory at 5200 Hex. It will be less confusing if it is all zeros, so if you have another program loaded there, you can turn off the computer and reload the DOS and DEBUG so that your display shows 64 bytes of zeros in memory.

Now type exactly the following, including the spaces:

M5200 AF 00 3C 3C 3D C3 01 52 X

Your screen should now display our program at 5200 Hex in memory. Here are the commands you used:

Mnnnn Load into memory at location nnnn
space bar Load next memory location
X Finished entering memory.

That is all it takes to enter a machine language program. If you made a mistake either use Mnnnn to change the mistake or start over at M5200.

You deserve another break. This time do not disturb the computer, but go to the kitchen and eat an apple, or take a trip to the powder room. When you come back we will play with our program.

Now we need to tell our computer where we put our program. To do this, enter RPC 5200. The R tells the monitor that we want to enter a value

into a register. PC tells the computer that the register we want is the Program Counter. The 5200 is the value that we want to put into the Program Counter. Now we have entered our program and told the computer where it is.

Let's see if our program works. Look at the contents of register A, displayed in the upper left corner of the screen. The first instruction is supposed to set this to 00. let's try it and find out if it works. The command I, for instruction, tells the computer to execute one instruction. Enter I and watch the contents of register A. Now look at register PC. It has also changed, and is now set to 5201, the address of the next instruction. Press I again. Register A should still read 00, and the Program Counter should now read 5201, because the instruction was a NOP or a "do nothing". Continue to single step through the program, and see if you understand what is happening.

After you have single stepped through the program several times, and understand what is happening by watching the program counter and register A, you are ready for another DEBUG command. This time enter G5200. This tells the computer to go to 5200 and execute the program there without stopping. What happens? Nothing happens? Well, it looks like nothing happens, but truthfully, a lot is happening. The computer is going through our little program over and over again, thousands of times each second, but nothing is displayed on the screen. If you want to check and make sure that the program is doing something press break and look at the updated display of the program counter and register A. Try typing G5200 and pressing break several times and observe the results. Now you know one way that you can get the computer to execute the program you enter.

Wouldn't it be nifty if we could get the program to execute as much as we wanted, and then stop at a particular spot so we could observe the state of the register? It would be nifty and we can do it.

Let's say we wanted to stop after the second increment instruction and before the decrement instruction. Enter G5200,5204. The comma and the second address are called the breakpoint. It tells the computer to stop when the program counter reads that number. Use the I instruction and look at the program counter to verify that the command does work.

Now wouldn't it be even niftier if we could get the computer to automatically stop after each instruction and update the display on the screen? Yes, it would, and we have a command for that. The command is U for update. What is not nifty is that in DOS Version 2.1, it does not work. They are trying to solve this problem, and promise something in version 3.0.

We have two more commands. C is like I, except that CALL instructions are executed in full. Right now I cannot take the space to explain more machine language programming, but you will be ready for it eventually if you continue to read this Journal. The other command is G and pressing enter. If you call the DEBUG routine from BASIC with CMD"D", this returns you to BASIC. Here is a table of the commands since you took that break to eat an apple.

R(register pair)nnnn -
Load indicated register pair with nnnn
I Execute single instruction
Gnnnn Go to memory location nnnn and execute the program there
Gnnnn,bbbb,cccc As above, but stop at bbbb and cccc
C Single instructions including calls
G(enter) Return to Basic
U Continously update display (DOESNOT WORK!)

Now I imagine that you are eager to get started analyzing the execution of all your basic programs. Theoretically, you know enough to do this. Unfortunately there is a fly in the ointment. For a few technical reasons, you cannot single step through BASIC programs successfully in DOS Version 2.1. This is another thing that is promised for Version 3.0.

If you would like to save your machine language program on diskette, use the DUMP command in the Disk Operating System. If you do so, you will have to write your programs above 6FFF Hex, as the DUMP command will not save programs below that address. To save a program you have written using the Mnnnn command, which starts at 7000 Hex and ends at 7OFF Hex, with execution beginning at 701A Hex and the file name "NAME" here is the format:

Press the reset button on the back of the keyboard beside the connector to the expansion interface. This will get you out of the DEBUG routine and load the DOS. When the computer prints out:

READY

type: DUMP NAME (START=X'7000',END=X'70FF',TRA=X'701A')

When you use this command, you must have a name with a blank before and after it, and the three addresses are the hexidecimal addresses of the first and last memory location in your program and the address at which the program is to begin. Use the above format exactly.

To verify the saving of your program, type DIR. It should be listed as NAME/CIM. To load it into memory, use LOAD"NAME/CIM".

If you have successfully completed this article and all the excercises, and if you have paid your tuition by subscribing to this Journal instead of borrowing it from a friend, you are now entitled to a degree. You may display after your name the initials DUMP, for Doctorate in Understanding Machine language Programming●

AUTOK · QEDIT

AUTOK and **QEDIT** make BASIC programming a breeze. **AUTOK** gives your keyboard auto-repeat: just hold any key down, and after a short delay the character repeats about eight times per second. **QEDIT** lets you edit any one-line BASIC statement on the screen, in place and in full view. It's much faster and easier to use than BASIC's EDIT; plus it even allows you to change line numbers. Included are a SYSTEM cassette for Level II, an instruction card, and special instructions for dumping the program to a DOS command file. **$15.**

The new SK version of **AUTOK/QEDIT** does all the above; plus it gives you single-keystroke entry of 25 common BASIC keywords. Just hold down the shift key, hit a letter key, and an entire word (FOR, NEXT, CHR$(, etc.) appears at once. The result? Greatly accelerated program entry. Version SK includes the Level II SYSTEM cassette, instruction card, DOS instructions, and a special keyboard layout card showing the command locations. **$19.**

WIN21

Learn to play expert blackjack with this practice and tutorial program. **WIN21** teaches you four effective strategies by Edward O. Thorp, author of **Beat the Dealer.** The computer is the house, and you bet, insure, split pairs, double down, etc. The program will coach you along the way, point out your mistakes, help you keep count, and keep track of your progress. This program is a must if you're planning a trip to Nevada or Atlantic City. **WIN21** includes a program cassette for 16K Level II, a copy of **Beat the Dealer,** and a comprehensive instruction booklet. **$29.**

LOAN · RETURN

LOAN is an amortization program which computes monthly payments, total principal, or loan duration given available data. It also computes the cost of a loan and displays a payment schedule. **RETURN** computes the internal rate of return on an investment, given the initial cost and dollar returns in subsequent years. This program is a must for businesses contemplating a major purchase. **LOAN** and **RETURN** on one cassette for 16K Level II, plus instructions: **$15.**

GOMOKU

With **GOMOKU** you play the computer in the ancient Japanese board game of five-in-a-row. Played on a 9x9 grid using full TRS-80 graphics, this game will provide countless hours of challenging excitement. Cassette for 16K Level II and instruction card: **$15.**

DISCOVERY BAY SOFTWARE CO.

P.O. Box 464 Port Townsend, WA 98368 Ph. (206) 385-4840

Name

Address

Zip

Please send:

_____ copies AUTOK/QEDIT @ $15 ea. $_____

_____ copies AUTOK/QEDIT (SK) @ $19 ea. _____

_____ copies LOAN/RETURN @ $15 ea. _____

_____ copies WIN21 @ $29 ea. _____

_____ copies GOMOKU @ $15 ea. _____

Subtotal $_____

Wash. residents add 5.1% sales tax: _____

TOTAL $_____

☐ Check/Money Order Enclosed.

☐ Charge VISA Card No. _____

☐ Charge Mastercharge* Exp. Date _____

*4 Digits Above Name _____

Signature Date

VISA®

BUSINESS SOFTWARE FROM THE PERIPHERAL PEOPLE

WE HONOR master charge THE INTERBANK CARD

Box 524

Mercer Island, WA. 98040

MAILROOM (32K disk) - A superb mail list program, permits review and edit, sorting by state of zip, with duplication checks. If file becomes too large, states can be separated and made new files. LPRINTS either labels or in tabular form. Entire file can be printed as well as specific states. Incoming disk files can be merged and saved as expanded files. $30.00 complete with documentation. *

THE ELECTRIC SECRETARY (16K disk) - One of the most powerful basic language word processing program ever written. Upper case only or U/L with conversion and suitable printer. Enter text such as letters, advertising, manuscripts or legal documents only once. The copy can be reviewed, revised, reworded, typos corrected or sentences added or deleted. Once the copy is edited, output to printer page formatted and justified camera ready copy. Program includes all software for upper/lower case, an echo routine to make your printer a typewriter, auto LF and even an electronic dictionary to hyphenate words. $75.00 complete with documentation. *

FREE - ***Upper/lower case, step by step conversion information complete with software routine. Complete kit of parts, plus cassette of Level II software. $20.00***

* On cassette or customer supplied DOS formatted disk.

PART II

Barber & Beauty Shop Cash Accounting & Payroll

By 80-U.S. Staff

(In the previous issue we covered the Data Accumulation, the DAILY CASH tape, the SUMMARY program and the CASH/GRAPH. Here, we will present the payroll program and sample runs of all the programs listed thus far.)

Before going into the payroll, let's take a look at Figure 1. This is a sample of what the screen will show when running the DAILY CASH program. After entering the information from the cash register tape, entering "30" as the operator number will clear the screen and display the total activity for the day. Then, when you have your daily cash tape in position, you may add that information to the tape.

Figure 2 represents a summary of two weeks. Note that as the data from the daily cash tape was read in, the date (if it fell between the dates asked for) was printed on the screen. Note also that the ending date is inclusive, the data for that date is also included in the summary.

THE PAYROLL PROGRAM

This program consists of three main parts. First, the payroll program itself, a section used to create a new pay data file on cassette and a section which allows auditing an already created pay data file on cassette.

Pay Data File Creation

Pay data file creation is selected by option 2 when this program is run. It takes place from line 2930 to line 3150.

Operator number/name and social security number need to be enclosed inside quotes so that you can use any type punctuation within them.

Most of the times entered here are universally understood, such as FICA and FEDTAX (we all know about them!). L & I refers to State Workman's Compensation, (in most states a portion of this is paid by the employee). PCMED is, in this case, a group medical plan into which the employee pays. MALP stands for malpractice insurance; again it represents the portion which the employee pays. RET RESERVE was explained in the first part of this article in the Mar-Apr 79 issue.

Auditing Pay Data Tapes

Auditing the pay data tapes is just what

```
                    S U M M A R Y

 FOR WEEK, PAYPERIOD, MONTH OR QUARTER - WITH
TOTALS FOR SERVICE/RETAIL, SALES TAX AND # CUSTOMERS

LOAD DAILY CASH TAPE AND SET TO READ
STARTING DATE,I.E.(MMDDYY)? 010179
ENDING DATE(INCLUSIVE)(MMDDYY)? 011279
 10279  10379  10479  10579  10679  10879  10979
 11079  11179  11279              ---------     S U M M A R Y

SUMMARY 10179 TO 11279
                SERVICE          RETAIL          #CUSTOMERS
OPERATOR 1       752             55.5             27
OPERATOR 2       618.5           29.7             29
OPERATOR 3       512.5           70.75            25
OPERATOR 4       157.5           0                15
OPERATOR 5       78              0                6
OPERATOR 6       776             69.2             38
OPERATOR 7       0               0                0

TOTALS           2894.5          225.15           140
TOTAL SER/RET=$ 3119.65          SALES TAX=$ 11.98
```

Figure 2

```
                D A I L Y   C A S H   P R O G R A M

THIS IS A DAILY CASH BREAKOUT PROGRAM
IT TAKES APART THE DAILY CASH REGISTER TAPE AND
SHOWS WHO DID WHAT FOR COMMISSION PURPOSES.

ENTER 30 AS OPERATOR # TO END PROGRAM

TODAYS DATE IS (MMDDYY)-? 011379
AMOUNT TAKEN FROM SHOP TODAY? 345.67
AMOUNT OF CHANGE LEFT YESTERDAY? 48.20
AMOUNT OF CHANGE LEFT TODAY? 54.68

OPERATOR #          ? 1
SER/RET 1/2         ? 1
AMOUNT              ? 18
OPERATOR #          ? 2
SER/RET 1/2         ? 1
AMOUNT              ? 34
OPERATOR #          ? 1
SER/RET 1/2         ? 2
AMOUNT              ? 5.95
OPERATOR #          ? 2
SER/RET 1/2         ? 1
AMOUNT              ? 6
OPERATOR #          ? 3
SER/RET 1/2         ? 1
AMOUNT              ? 12
OPERATOR #          ? 2
SER/RET 1/2         ? 1
AMOUNT              ? 38.5
OPERATOR #          ? 30


011379
                SERVICE         RETAIL          # OF CUSTOMERS
OP 1             18              5.95           1
OP 2             78.5            0              3
OP 3             12              0              1
OP 4             0               0              0
OP 5             0               0              0
OP 6             0               0              0
OP 7             0               0              0

SERVICE          RETAIL          SALES          TOTAL SERVICE
TOTAL            TOTAL           TAX            PLUS RETAIL
 108.5            5.95            .30345         114.45
TOTAL SERVICE CUSTOMERS FOR TODAY = 5
CASH IS OVER TAPE BY 237.397

POSITION DAILY CASH TAPE TO RECORD & ENTER?
```

Figure 1

it says: you can load any employee pay data tape and view it on the screen (using option 3). See figure 5.

The Payroll Program

The payroll program (option 1) asks for dates from the DAILY CASH tape, almost in the same way that the SUMMARY program did. Data from the DAILY CASH tape is accumulated for the period requested for ALL employees in lines 2160 through 2290.

The data being read from the DAILY CASH tape is gross intake for each employee. Some means must be provided to calculate each individual commission, and this is done in the subroutine at 2860.

The subroutine at 2860 first asks if the service gross was in excess of $50.00. If it was, then the commission is figured at one half of 50, times 55% of everything above 50 dollars. If the service gross was less than $50.00, line 2880 is ignored, and the program asks if it was greater than 37 dollars. If it was, the commission is figured at one half. If service gross is less than 37 dollars, line 2890 is ignored and line 2900 then asks if it was greater than $18.50, in which case the commission is $18.50 since this is the guaranteed minimum. Line 2910 checks to see that anyone making less than $18.50 still receives the minimum.

Since all retail commission (in our example) is 10% of retail price, line 2870 adds this in for each employee. Line 2860 says that if an employee did not sell any retail products or do any service work, they must not have been at work that day. It then gives a RETURN, without adding the guaranteed minimum to that employee. (Be careful here if your employee comes in on their day off and buys some product using their own operator number!!)

Once the DAILY CASH tape is read in, all the amounts for all the operators are in memory (in variables MM thru ZZ, see lines 2410 through 2470). Using the command "RUN" after this point will wipe out all the variables, so use caution! (you should not really have to use it).

See Figure 3; you should now be at the point where the program asks you to ENTER OPERATOR NUMBER?

You can do the payroll in any order, but if you tell it to do Operator 1 and read in the tape for Operator 2, line 2400 will catch your mistake and give you another chance.

Immediately after reading in the pay data tape for an employee, the screen will clear and the display will show the pay stub (see figure 4). You can then copy the information from the screen, screen print it if you have one, or change

```
                    PAYROLL PROGRAMS

        -1- PAYROLL

        -2- CREATE OPERATOR DATA FILES

        -3- AUDIT OPERATOR DATA FILES


        TYPE 1, 2, OR 3 AND ENTER? 1
                              P A Y R O L L

        USES DAILY CASH TAPE AND OPERATOR DATA TAPE.
        STARTING DATE, IE (MMDDYY)? 010179
        ENDING DATE (INCLUSIVE)(MMDDYY)? 011279

        LOAD DAILY CASH TAPE, POSITION AND ENTER?
         10279  10379  10479  10579  10679  10879  10979  11079
          11179  11279 DATA INPUT IS NOW COMPLETE
        ENTER OPERATOR NUMBER? 2
        ENTER OPERATOR # 2 HOURS WORKED THIS PERIOD
        ? 80
        LOAD OP # 2 DATA TAPE AND PRESS ENTER
        ?
```

Figure 3

```
2 BETTY BEAUTICIAN 123-45-6789  PERIOD ENDING      11279

HOURS        SER GROSS      RETAIL GROSS   TOTAL        NET PAY
 80           325.18           2.97         328.15       242.97

             FICA     FEDTAX      L&I     PCM     MALP    TOT DED
CURRENT      19.85    45.94       1.46    14.2    .75     82.21
YEAR TO
DATE         142.95   282.72      5.69    33      4.25    280.97
RETAIL RESERVE IS NOW= $ 37.47
```

Figure 4

```
                PAYROLL PROGRAMS

-1- PAYROLL

-2- CREATE PAY DATA FILES

-3- AUDIT PAY DATA FILES


TYPE 1, 2, OR 3 AND ENTER? 3
     A U D I T  P A Y  D A T A  F I L E

ALLOWS YOU TO CHECK DATA IN FILES

INSERT PAY DATA TAPE AND SET TO READ
PUSH ENTER WHEN READY?
OP#, NAME ------------2 BETTY BEAUTICIAN
SOCIAL SECURITY # -----123-45-6789
FED TAX PERCENTAGE ---- .14
PCMED DEDUCTION ------- 14.2
YEAR TO DATE FICA ----- 123.1
YEAR TO DATE FED TAX -- 236.78
YEAR TO DATE L & I ---- 4.23
YEAR TO DATE PC MED --- 18.8
YEAR TO DATE MALP ----- 3.5
YEAR TO DATE TOT DED -- 198.76
YEAR TO DATE RET RES -- 34.5


WANNA DO IT AGAIN ? 1 YES 2 NO? 2
```

Figure 5

the PRINTS to LPRINT if you have a line printer.

The tax liability figure is printed after each pay stub is printed, and accumulates for the whole session. This is for information only, and gives some idea to the employer about how much to set aside for the tax collector. As with most of the figures in these programs, you may have to change the constants to fit your particular local and state governmental requirements.

After this, you can push enter and the program tells you to rewind operator "X" tape and set to record. This allows you to use the same tape over again, with the updated information on it for the next pay period. After this, pressing enter to continue will get you to the next employee.

CONCLUSION

As we mentioned in the first part of this article, it is unlikely that many can use this program "as is". it would be nice if everyone had the same bookkeeping system, but that would be rather dull too.

The intent is to provide information on one way to do the job, with the user adapting it to their particular need and changing it to fit their specific requirements.

There are many things, as always, which can be done to enhance and improve this type of program. Tape systems in general are bulky and slow to use. One suggested way to improve it would be to use two tape recorders, with the old information on one and the new on the other.

If you are limited to tape operation, this may be a starting point from which you may proceed on your own ●

```
2000 CLS
2010 PRINT"                    PAYROLL PROGRAMS"
2020 PRINT:PRINT:PRINT"-1- PAYROLL"
2030 PRINT:PRINT"-2- CREATE OPERATOR DATA FILES"
2040 PRINT:PRINT"-3- AUDIT OPERATOR DATA FILES"
2050 PRINT:PRINT:INPUT"TYPE 1, 2, OR 3 AND ENTER";D2
2060 ON D2 GOTO 2070,2930,3160
2070 CLS
2080 PRINT"                          P A Y R O L L"
2090 PRINT:PRINT"USES DAILY CASH TAPE AND OPERATOR DATA TAPE.
2100 INPUT"STARTING DATE, IE (MMDDYY)";A1
2110 INPUT"ENDING DATE (INCLUSIVE)(MMDDYY)";A2
2120 PRINT:INPUT"LOAD DAILY CASH TAPE, POSITION AND ENTER";
2130 INPUT#-1,A$,E,F,G,H,I,J,K,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
2140 A3=VAL(A$)
2150 IFA3<A1GOTO2130
2160 A=M:B=N:GOSUB 2860
2170 MM=MM+C:NN=NN+D
2180 A=O:B=P:GOSUB 2860
2190 OO=OO+C:PP=PP+D
2200 A=Q:B=R:GOSUB 2860
2210 QQ=QQ+C:RR=RR+D
2220 A=S:B=T:GOSUB 2860
2230 SS=SS+C:TT=TT+D
2240 A=U:B=V:GOSUB 2860
2250 UU=UU+C:VV=VV+D
```

Note:
0 = zero
O = oh

```
2260 A=W:B=X:GOSUB 2860
2270 WW=WW+C:XX=XX+D
2280 A=Y:B=Z:GOSUB2860
2290 YY=YY+C:ZZ=ZZ+D
2300 A=0:B=0:C=0:D=0
2310 PRINTA3;
2320 IFA3>=A2GOTO2350
2330 GOTO2130
2340 REM - HERE IS WHERE YOU INPUT OPERATOR DATA TAPES
2350 PRINT"DATA INPUT IS NOW COMPLETE"
2360 INPUT"ENTER OPERATOR NUMBER";F1
2370 PRINT"ENTER OPERATOR #";F1;"HOURS WORKED THIS PERIOD":INPUT H1
2380 PRINT"LOAD OP #";F1;"DATA TAPE AND PRESS ENTER":INPUT;
2390 INPUT#-1,A$,B$,H2,H3,O1,O2,O3,O4,O5,O6,O7
2400 IF VAL(A$)<>F1 THEN CLS:PRINT"REDO":GOTO2370
2410 IFF1=1THENSG=MM:RG=NN
2420 IFF1=2THENSG=OO:RG=PP
2430 IFF1=3THENSG=QQ:RG=RR
2440 IFF1=4THENSG=SS:RG=TT
2450 IFF1=5THENSG=UU:RG=VV
2460 IFF1=6THENSG=WW:RG=XX
2470 IFF1=7THENSG=YY:RG=ZZ
2480 MA=.75:LI=.0183*H1:GP=SG+RG:CF=GP*.0605:CT=GP*H2
2490 IFGP=0THENMA=0
2500 TD=CT+CF+LI+MA+H3:NP=SG-TD
2510 O1=O1+CF:O2=O2+CT:O3=O3+LI:O4=O4+H3:O5=O5+MA:O6=O6+TD:O7=O7+RG
2520 CLS:PRINTTAB(1)A$;TAB(20)B$;TAB(33)"PERIOD ENDING";TAB(51)A2
2530 SG=INT(SG*100+.5)/100:RG=INT(RG*100+.5)/100
2540 GP=INT(GP*100+.5)/100:NP=INT(NP*100+.5)/100
2550 PRINT
2560 PRINTTAB(1)"HOURS";TAB(14)"SER GROSS";TAB(28)"RETAIL GROSS";
2570 PRINTTAB(42)"TOTAL";TAB(54)"NET PAY"
2580 PRINTTAB(1)H1;TAB(14)SG;TAB(28)RG;TAB(42)GP;TAB(54)NP
2590 PRINT
2600 PRINTTAB(12)"FICA";TAB(20)"FEDTAX";TAB(32)"L&I";TAB(39)"PCM";
2610 PRINTTAB(47)"MALP";TAB(54)"TOT DED"
2620 CF=INT(CF*100+.5)/100:CT=INT(CT*100+.5)/100
2630 LI=INT(LI*100+.5)/100:TD=INT(TD*100+.5)/100
2640 PRINT"CURRENT";TAB(11)CF;TAB(19)CT;TAB(31)LI;TAB(39)H3;
2650 PRINTTAB(47)MA;TAB(54)TD
2660 O1=INT(O1*100+.5)/100:O2=INT(O2*100+.5)/100
2670 O3=INT(O3*100+.5)/100:O4=INT(O4*100+.5)/100
2680 O5=INT(O5*100+.5)/100:O6=INT(O6*100+.5)/100
2690 O7=INT(O7*100+.5)/100
2700 PRINT"YEAR TO"
2710 PRINT"DATE";TAB(11)O1;TAB(19)O2;TAB(31)O3;TAB(39)O4;
2720 PRINTTAB(47)O5;TAB(54)O6
2730 PRINT
2740 PRINT"RETAIL RESERVE IS NOW= $";O7
2750 TL=TL+(CF*2)+CT+(GP*.037)+(H1*.0595)+((SG/.55)*.0106)+((RG/.1)*.056)
2760 TL=INT(TL*100+.5)/100
2770 PRINT"TAX LIABILITY FOR THIS PERIOD = $";TL
2780 INPUT"FOR RECORDING INSTR, PRESS ENTER";
2790 PRINT"REWIND OP #";F1;"TAPE AND SET TO RECORD"
2800 INPUT"WHEN READY, PUSH ENTER";
```

```
2810 PRINT#-1,A$,B$,H2,H3,O1,O2,O3,O4,O5,O6,O7
2820 PRINT"RECORD COMPLETE, REWIND AND REMOVE OP#";F1;"TAPE"
2830 INPUT"PRESS ENTER TO CONTINUE";
2840 CLS:GOTO2360
2850 END:'FOL IS % ROUTINE FOR DAILY OPERATOR GROSS
2860 IFA<=0 AND B<=0 THEN C=0:D=0:RETURN
2870 D=.1*B
2880 IFA>50THENC=25+.55*(A-50):RETURN
2890 IFA>37THENC=A/2:RETURN
2900 IFA>18.5THENC=18.5:RETURN
2910 IFA<18.5THENC=18.5:RETURN
2920 END 'UPDATE OR CREATE NEW OPERATOR DATA FILE
2930 CLS:PRINT"    P A Y  D A T A  F I L E  C R E A T I O N"
2940 PRINT:PRINT"FOLLOW THE PROMPTS TO CREATE AN OPERATOR DATA FILE"
2950 PRINT"ENCLOSE OP#NAME,SS# IN QUOTES"
2960 PRINT"ENTER A 0 FOR THOSE ITEMS WHICH DO NOT APPLY"
2970 INPUT"OP#,NAME,IE,(3 MARY DOE)";A$
2980 INPUT"SOCIAL SECURITY #";B$
2990 INPUT"FED TAX % (.12)";H2
3000 INPUT"MED INSUR PER PERIOD";H3
3010 INPUT"OLD FICA";O1
3020 INPUT"OLD FEDTAX";O2
3030 INPUT"OLD L&I";O3
3040 INPUT"OLD PCMED";O4
3050 INPUT"OLD MALP";O5
3060 INPUT"OLD TOT DED";O6
3070 INPUT"OLD RET RESERVE";O7
3080 PRINT"POSITION A CLEAN TAPE, MARKED WITH THE OPERATOR #,"
3090 PRINT"IN RECORDER AND SET TO RECORD"
3100 INPUT"WHEN READY TO RECORD, PRESS ENTER";
3110 PRINT#-1,A$,B$,H2,H3,O1,O2,O3,O4,O5,O6,O7
3120 PRINT"RECORD COMPLETE"
3130 INPUT"DO YOU WANT TO MAKE MORE 1 YES 2 NO";D1
3140 IFD1=2THEN STOP
3150 IFD1=1THEN CLS:RUN2000
3160 CLS:PRINT"     A U D I T  P A Y  D A T A  F I L E"
3170 PRINT:PRINT"ALLOWS YOU TO CHECK OPERATOR DATA IN FILES"
3180 PRINT:PRINT"INSERT OPERATOR DATA TAPE AND SET TO READ"
3190 INPUT"PUSH ENTER WHEN READY";
3200 INPUT#-1,A$,B$,H2,H3,O1,O2,O3,O4,O5,O6,O7
3210 CLS
3220 PRINT"OP#, NAME ----------";A$
3230 PRINT"SOCIAL SECURITY # -----";B$
3240 PRINT"FED TAX PERCENTAGE ----";H2
3250 PRINT"PCMED DEDUCTION -------";H3
3260 PRINT"YEAR TO DATE FICA -----";O1
3270 PRINT"YEAR TO DATE FED TAX --";O2
3280 PRINT"YEAR TO DATE L & I ----";O3
3290 PRINT"YEAR TO DATE PC MED ---";O4
3300 PRINT"YEAR TO DATE MALP -----";O5
3310 PRINT"YEAR TO DATE TOT DED --";O6
3320 PRINT"YEAR TO DATE RET RES --";O7
3330 PRINT:PRINT
3340 INPUT"WANNA DO IT AGAIN ? 1 YES 2 NO";D3
3350 IFD3=1GOTO3160ELSE IF D3=2 RUN 2000
```

TRS80 FORTRAN *an evaluation*

In the last issue we covered the TRS80 FORTRAN package and discussed what features it provided ("NOW THE TRS-80 HAS FORTRAN", Mar-Apr 79). In this issue, we will cover a month's experience with the FORTRAN package and try to let you know whether this is the right purchase to extend your system.

By T. Dettman, Associate Editor

FIRST IMPRESSIONS

When I first received the FORTRAN package and tore into the manual provided by Microsoft, I was impressed with the manual as a collection of information on the system they provided. You actually receive four manuals in one with the package. The first is a TRS80 User's Manual which describes how the system works on the TRS80 as well as how to use the compiler.

My first step on opening the manual was to follow through the step by step instructions for inputting a simple FORTRAN program (a listing is provided for a program to make a table of Fahrenheit to Centigrade conversions). Despite a minor error in the listing (which has now been corrected in the manuals), I was able to get the program running exactly as specified.

This test introduced me, though, to the most frustrating aspect of working with the system, and that is the time involved to get a program running satisfactorily. To run a program, you must go through the following steps:

1. Load the Editor to create the program file.
2. Store the program file on disk.
3. Change diskettes and load the compiler (if you have three disk drives you can have both diskettes and the program disk in at the same time).
4. Compile the program and store the machine code on disk (the manual suggests compiling first without generating the machine code to be sure there are no syntax errors).
5. Reload the diskette the edior was on, and load the linking loader.
6. Load and execute the program.

If you are lucky, at this point, you can now save a working machine language program on disk as a command file for execution directly from DOS. In general, you won't be so lucky and you will have a small error, "a bug" in the program. When you do you will have to find it, correct it, and then go through all of the steps above again!

As you can see, correcting errors in the same way that most people do, with an interpreter, by changing a statement and rerunning the program, will just lead you to throw the system out the window and swear off FORTRAN for good. Don't do it!

The amount of time it takes to program and debug a FORTRAN program with this system is well compensated for by the ability to generate machine language programs at the end of the line. For programs that will be used repeatedly, this capability, will in the long run, save more time than you will lose if you are a careful programmer. How can you minimize the time that it takes you to program in FORTRAN and Debug? The answer lies in a complete knowledge of the second manual, the FORTRAN-80 reference manual.

THE FORTRAN-80 REFERENCE MANUAL

Anyone with experience on the 8080 version of the Microsoft FORTRAN package will recognize this manual. It is the same manual used for all of their 8080 and Z80 basedFORTRAN packages.

The manual is most definitely not intended to be a text on the FORTRAN language as Microsoft very clearly points out. However, it is a concise description of what is contained in the FORTRAN language in the package (it is similar to the manual provided with Level II BASIC in terms of its coverage and its intent).

THE UTILITY SOFTWARE MANUAL

The Utility Software Manual covers the MACRO-80 assembler and the linking loader. Like the FORTRAN manual it is not a text on assembly language, but it does give the features available with the package. It is important to read the loader manual carefully so that you can make best use of the loader's capabilities.

THE EDIT-80 USER'S GUIDE

The last manual with the package is the EDIT-80 user's guide. This one manual should be digested early or you will continually make trivial editing errors that could have been avoided. As an example, I have had a tendency to type "END" instead of "E" when exiting from the editor. Since the program looks only for the first letter as a command and interprets everything else as the file name to store the program under, I have many files generated called "ND".

THE PROGRAMS

Let's look more closely at the programs provided to see what you can and cannot accomplish with the system. First, we'll look at the comiler and the FORTRAN language itself.

THE F80 COMPILER

The heart of the TRS80 FORTRAN

package is the compiler which actually generates the machine language code corresponding to the FORTRAN statements you program. To test the system, I took two programs from my FORTRAN library, one written originally for an IBM system, to do simple statistics such as average, mean, etc., and the other from a CDC system written originally to do a Fast Fourier Transform.

Both programs worked well after an initial debugging period in which I had to spend a great deal of time reading the manual to find the little problems that plagued the conversion. Some specific problems involved the use of the DIMENSION statement to initialize an array in a subroutine. A standard practice is to set the array dimensions to '1' in the subroutine and pass the actual dimensions only in execution. On large computers this will work and provide a flexible subroutine, not limited in array size before use. This practice led to errors in the program during execution. The compiler did not identify them.

The FORTRAN manual does allow you to use variable array dimensions if the dimensions are passed explicitly in the subroutine calling statement. For example:

```
      DIMENSION A(3,4,5)
      .
      .
      CALL TEST(AZ,3,4,5)
      .
      .
      SUBROUTINE TEST(X,I,J,K)
        DIMENSION X (I,J,K)
```

This type of capability gives you the flexibility to write general subroutines for many situations that can easily be loaded as needed for a problem.

Aside from the standard features of ANSI FORTRAN, the Microsoft system also provides:

1. A STOP or a PAUSE statement may be followed by a six character ASCII string that will be displayed on the screen when the statement is executed
2. WRITE statements may have an EOF and/or ERROR branch included in the program statement to transfer control to an appropriate part of the program
3. PEEK, POKE, INP, and OUT functions have been added to the function library
4. FUNCTION definitions may include the use of subscripted variables
5. HEX constants may be used
6. String constants may be written with apostrophes as delimiters
7. Strings are allowed in expressions in place of integer constants
8. Statements may be carried over to any number of continuation lines
9. Expressions may use mixed mode (integer & real for example) arithmetic with automatic conversion

The only restriction that I had to get around was the inability to handle complex numbers (required for the Fast Fourier Transform).

The F80 Compiler itself is accessed as a /CMD file in DOS by typing in "F80". Commands to the compiler are then given by using the form:

OOOO , LLLL = SSSS -SWITCHES

The command format allows you to specify the filename where the object code is to be written (OOOO), the file name the listing of the compiled program is to be written (LLLL) as well as the name of the file to be compiled (SSSS). The switches appended to the end allow you to specify various compiling options:

1. Print listing addresses in OCTAL
2. Print listing addresses in HEX
3. List only the FORTRAN statements and not the compiled assembly language code
4. Add extra stack space for the compilation
5. Format the compiled code for loading to a ROM

The minimum command is simply "=SSSS" which will compile a program without generating either a listing or the object code. This is useful for checking the syntax.

One of the major advantages of the compiler is that it generates fully relocatable code for the FORTRAN program which can then be loaded by the Linking Loader in combination with other relocatable modules at whatever location is desired.

One of the nicest features of the compiler is the listing produced in the compilation process. It will list the full Z80 code for the program as generated in Z80 mnemonics. This can be a useful device for helping to learn assembly language programming or for program debugging. Each FORTRAN statement is broken down to a series of Z80 mnemonics and listed with the source code on a file when called for with the LLLL filename. To see the file, one need only type LIST LLLL in DOS to list it on the screen. PRINT LLLL will cause the file to be printed on the line printer.

THE LINK80 LINKING LOADER

The Link80 Loader takes relocatable modules from the Fortran Compiler, the MAC80 assembler, the FORTRAN library (FORLIB), or a user generated subroutine library and puts them together as an executable unit. It can also store the program on disk as a /CMD file.

Switches are provided in the syntax of the loader to perform the following tasks:

1. Start execution immediately after loading
2. Save the program on disk
3. Set the location of the program in memory
4. List the starting and ending points of the program and data storage areas
5. Search a specified library for subroutines

THE EDIT80 EDITOR

The really fundamental program is the Editor. No package of programs such as this can achieve its optimum performance if the editor is incapable of conveniently handling the source code. I am happy to report that the editor is in fact capable of bringing out the most that the system can offer. Many commands will be familiar to users of Level II editing or Editor Assembler editing. However, this very familiarity can lead to troubles, since many of the conventions differ slightly.

On entering the editor, you are editing in a line mode that is suitable for adding or deleting whole lines in the source program. In order to edit individual characters, it is necessary to enter the "Alter" mode for a given line. Once in the Alter mode, the commands are similar to those in BASIC editor, but there are powerful extensions that will make editing far easier for those willing to learn the differences.

In the line editing mode, lines can be inserted, deleted, replaced, printed, listed or renumbered. In the character editing mode, insertion, deletion, replacement, as well as extension are possible. In both modes, it is also possible to search for a given string and/or replace that string alone. This is a capability found generallly on the text editors for large systems and is extremely powerful when used correctly.

The editor also has several interesting switch options which extend the utility of the program. With the BASIC switch, a BASIC program saved with the 'A' option on disk may be read. The BASIC line numbers will be used as the editor sequence numbers. (Don't try to renumber the program since the editor will not check line number references in GOTO's, GOSUB's, etc.) A file can be saved as an ASCII file by specifying the "UNSEQ" switch. It the switch is set for a program that was read in with the BASIC switch, it will be written with the line numbers in ASCII.

The "SEQ" switch allows reading in a file as if the BASIC switch were used, but on completion, it will be written to disk in standard editor format.

SOME RANDOM NOTES ABOUT THE SYSTEM AND ITS USE

For small programs such as the test program shown, the system is too large and unwieldy. To demonstrate this I ran the following program on the FORTRAN system:

```
          WRITE(1,100)
100   FORMAT('TEST')
          END
```

Starting a stop watch when I pressed

the ENTER key to start the editor, it took 3 minutes 41 seconds to get the program through the editor, compiler, and loader and execute it. After another minute and a half, I had put the program on disk as a /CMD file and could execute it from DOS by typing the name of the file.

For comparison, I typed in the following BASIC program to accomplish the same task:

```
10 PRINT "TEST"
```

I was able to type this program in and RUN it in 14 seconds even allowing for rereading to make sure I didn't make an error and to eat up time!

This trivial test points out that the FORTRAN package is not for quick calculations. The BASIC interpreter in Level II or Disk BASIC is far more suitable for immediate calculations or for testing out and proving algorithms. They will also make debugging more rapid since you have to go through roughly four minutes of running programs in the FORTRAN package in order to make only one run of a program. Immediate corrections are nearly impossible.

The FORTRAN package also sufferes from the many problems of standard FORTRAN. In particular string handling is inconvenient compared to Level II BASIC. In order to do string operations, you have to program around a language that is intended for algebraic computations with numbers.

A more particular problem with the FORTRAN package is that the language itself is inherently more flexible, and hence more prone to human error than BASIC. FORTRAN has always given a large amount of flexibility in programming but has required that the programmer be careful on a step by step basis with his coding. Each detail must be looked at carefully and its affect on the program assessed. With the large time overhead for compiling and loading, the need for care is even more acute.

Due to the limited size of a microcomputer system, the error messages available from the compiler and for run time are less than most FORTRAN programmers are used to. On large systems, the FORTRAN compiler typically will check literally hundreds of fatal error conditions and will even suggest that you are programming acceptably but not efficiently. It may even check for non-standard use of the FORTRAN language in terms of the ANSI standard.

The Microsoft FORTRAN provides 35 Fatal and 33 Warning error messages. These can at times be obscure and may be wholly inadequate. During program execution, 14 Warning and 11 Fatal errors are avaiable. All of these errors are of necessity limited. More obscure errors will show up, if at all, as one of these standard error messages. For example, an error in execution could wipe out a portion of memory and leave you with an error that indicates that a statement executed improperly. This might not be true, the statement executing improperly could have been caused by the way you stored data (improperly dimensioned variable arrays and so forth.)

CONCLUDING NOTE

One month's experience with the FORTRAN package has made me a devotee. I have always liked the FORTRAN language anyway, and the capability of programming in it on my TRS80 is great!

With Microsoft's package, I can generate a library of my frequently used subroutines in FORTRAN from routines I already have, to do a myriad of jobs. Further, I can make machine language programs that will execute directly from disk and that will run on any TRS80, not necessarily my own. The possibilities are limitless since I can generate machine language files without going to assembly language if I don't want to.

If you feel that I am enthusiastic about the package, you are right. But that enthusiasm is tempered by the fact that I realize the package is not for every TRS80 user. Unless you are either already familiar with the language or you seriously intend to learn it, this could be a wasted purchase. Even if you do know FORTRAN, you must want to have the capabilities of the package enough to be willing to put up with the limitations, most particularly with the time required for each run. If you buy the system without realizing that FORTRAN itself will demand that you practice *good* programming practices instead of the 'cut and try' technique so often used on BASIC interpreters, you will be fed up with it after you have run your program for two days and still haven't figured out the error messages.

Engineers and scientists who use their TRS80's will probably want this package since their backgrounds in programming are generally FORTRAN based. Other users will find that there can be great benefit in learning the FORTRAN language in order to get the flexibility it provides and the compiler.

80-U.S. will publish a series of tutorial articles on FORTRAN, based on the Microsoft FORTRAN compiler to help those who would like to learn the FORTRAN language. Watch for the start in the next issue.

SPECIAL NOTE

Microsoft has now started offering the MAC80 assembler in a package with the Linking Loader and the Editor. How good is the Assembler? How does it compare with the TRS80 Editor Assembler package? Are the two assemblers compatible with each other? These and many other questions will be covered in the next issue when 80-U.S. evaluates the MAC80 assembler ●

A note on disk basic

If you work with large disk files, you might run into the situation where you are outputting to disk with a PRINT# statement, and you get a DISK FULL ERROR. It's not really a probelm (other than an inconvenience), but you have to realize at the beginning that the last record to disk was never written! Why, you ask? Well, it's simple, the disk operating system does not write directly to disk but writes to that 256 byte buffer you established when you answer the number of files question on entering DISK BASIC. A write to disk is only done when the buffer is full. Before this write however, the disk full condition occurs to prevent another write and the data sits in memory. If you still have the data in memory, you can recover it, but don't expect it to be on the disk, it won't be.

Ever decide that you would like to keep some of those programs you use frequently as subroutines and load them only when need in programs you are writing, but you would rather not have to type them in each time? Well, your problem is solved in DISK BASIC if you use the MERGE command. Here's how to create your own subroutine library on disk.:

1. Type each subroutine into your computer and save them on disk as ASCII files. To do this, save them with the command:

```
SAVE "filename",A
```

2. After your disk library is complete, type your main program as if the subroutine were there

3. After the program is complete, add any subroutines you need with the command:

```
MERGE "filename"
```

This will cause the subroutine to be added to the program in memory as if you had typed it yourself. You have to be careful that the subroutines use statement numbers that are not used in your main program or in other subroutines. if you have statement numbers in the main program that are in the subroutine, they will be replaced with the subroutine's statement numbers. For a subroutine file, try to keep all subroutines with separate numbers and start them with statement numbers over 30000. In this way you can append subroutines to your programs as needed once they are properly checked out.

UNCLASSIFIED ADS

SCREEN PRINTER (used only 10 days) with original paper and cable. First $375.00 takes it. Send address (no cash) and phone. Logics One, PO Box 41, Utica, Michigan 48087 (I will pay shipping).

TRS-80 YELLOW PAGE on hardware & software. Send $1.00 plus 2 SASE for 2 issues. Vendors send $20.00 & product description. 96 Dothan St., Arlington, MA 02174.

TRADE-TRADE-TRADE. I have over 300 quality TRS-80 programs on DISK. Both machine language and Basic. Business programs, Games, Application, Education. Send me your disk with 10 or more of your best and I'll send you back 10 or more of my best. Bill Vick. 3325 Landershire Lane. Plano, Texas 75023 (214) 596-0533.

FOR SALE- NEW CENTRONICS P-1 PRINTER Identical to RS Quick Printer. With 4 rolls of paper. $400. C H Gallion. 6544 S. Humphrey Rd. Clinton, WA 98236 (206) 321-4226.

TRS-80 L2 ORIGINAL SOFTWARE. Music, Magic & of course Games. Keyboard: play melodies on your TRS-80 Card Trick: computer guesses card(s) chosen by up to 3 people. 5 Speed Racer: Real time action. Arithmetic Drill: Giant numerals & right to left entry + - * / $5.00 each or all 4 for $15.00 SASE for complete list to Advanced Graphics 920 W Romneya #6. Anaheim, CA 92801

WE HAVE AN RS 232 serial interface, plug compatable with TRS-80 Exp Interface with software & instr. Precision Electronics, 3411 S 90th St. Tacoma, WA 98409.

OWNERS or USERS of TRS-80 L2 or DOS systems interested in doing community/humanitarian service drop a postcard with name, address & type equip to Mike Freeman, 946 Alder St. Tacoma, WA 98406.

STEVEN'S SOFTWARE has the disk oriented Text Pencil! A few of the features are right justification, search for word or phrase and variable columns. $20.00 on formatted diskette, $18.00 on cassette tape, $15.00 on user supplied DOS, Steven's Software, 9005 East Vickery Rd., Tacoma, WA 98446.

TRADER VICK... Over 300 quality programs on disk to trade or swap for software or hardware. What do you have? Trader Vick, 3325 Landershire Lane, Plano, TX, 75023 (214)596-0533.

80-U.S.Journal Unclassified Rates

$2.50 per half inch and $2.50 for each additional half inch per insertion. Send printed, or typed ad with correct amount to 80-U.S. Journal, P.O. Box 7112, Tacoma, WA. 98407.

TAKE THE TIME TO WRITE UP THAT AD!

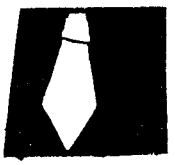
IT WILL BE READ- GET FAST ACTION!

TRS-80, & PET MICRO COMPUTERS

This is a warning! IF your OWNER doesn't deliver A #10 SASE for further instructions. You WILL Miss the GREATEST opportunity to run virtually every Program available free THIS IS YOUR last CHANCE! - The WriTers are GETTING NERVOUS.

TRENDY IV, THE cassette MOB inc.

COMP. SERV. DIV.
dept. 1230
P.O.Box 7433 Hollywood,
Florida 33021



---

* NEW * WARSIM * NEW *

A TWO-PLAYER WARGAMING SYSTEM THAT SIMULATES THE ACTION OF TWO COMBAT SHIPS ON A SEA WITH UP TO 6 PORTS, 5 SHALLOWS AND 5 MINEFIELDS.

SYSTEM IS AN EVENT DRIVEN SIMULATOR THAT SCHEDULES AND EXECUTES 12 EVENTS THAT CONTROL THE SYSTEM'S 18 ENTITIES AND 182 ATTRIBUTES.

ALL RELEVANT ENTITIES AND ATTRIBUTES ARE USER DEFINABLE IN DATA STATEMENTS, ALLOWING FOR AN UNLIMITED NUMBER OF GAMES FROM THE SIMPLE TO THE MOST COMPLEX.

SYSTEM WRITTEN IN LEVEL II BASIC USING 16K.

PRICE OF SYSTEM INCLUDES CASSETTE, USER MANUAL, POSTAGE AND HANDLING.

SEND $19.95 (CHECK OR MONEY ORDER) TO:

JORDAN DATA PROCESSING
4851 SUMMIT HILLS WAY
TUCKER, GA. 30084

Business Computing

The purpose of this article is: To provide a better understanding of computers and computer software and to emphasize that a computer will not immediately replace your bookkeeper, CPA, and five employees...or that it will immediately solve all your business and accounting problems.

By John Strader, Lacey, Wa

The majority of the readers appear to be in the same situation I was when I bought my TRS-80 system: interested in learning about computers and/or using the computer in a business-occupation. Some of you may have found that the TRS-80 will not work in your business because of a lack of available software, or the disk files are not large enough to handle the volume of information in your business. For others the system works very well because good software suited to the machine is available. Be patient; time will probably solve both problems. It usually takes a minimum of six months to a year to develop a good software application's package.

In a business the ultimate goal should be to provide your business with a management information system. The system will vary from business to business. The initial action should be to conduct a survey to determine your business needs. The survey should answer the following questions: How much disk space is needed to hold the information? What type of applications are suitable for computerization? Will the speed of the computer/printer and disk allow you to enter the necessary volume of information?

To illustrate, I will list what the most complete system would be for the average small to medium size retail business, which requires a hard copy invoice to the customer, sells on account, and rings up sales in a cash register. (Note: some businesses may only need a portion of this system because of competitive conditions, methods or operation, or cost-benefit relationships.)

1. A perpetual inventory system with an economic order quantity reorder system.
2. An invoice billing system and accounts receivable package with an aging routine.
3. A check and cash disbursement package.
4. A program with optical scanner, bar code reader, or voice input to enter information into this system.
5. A detailed cost system; this system should be interfaced to the inventory package and provide for estimating the cost of income from various lines of merchandise or individual items.
6. A payroll system which may or may not interface to the cost system depending on the business's needs.
7. A general ledger-journal package which will accept entry from the programs listed above.
8. Various special programs depending on the type of business, i.e., for a pharmacist to prepare prescriptions, print labels, prepare his prescription records and fill out the necessary government paper work.
9. Some type of text editing system.

To my knowledge, there is nothing available from the main frame manufactureres which includes the entire system above at a price the average small to medium size business could afford (under $15,000 for the hardware and software). The majority of the present microcomputers are not capable of operating the entire system listed above because of inadequate disk storage, slow disk access time, use interpreters rather than compiled applications software. There are systems which have some or many of the attributes listed above. Generally a microcomputer user must be satisfied with a part of the system unless he/she is a lawyer or accountant.

Generally a decision must be made to either postpone the use of the computer until the necessary systems become available or use the microcomputer as an intelligent terminal in a timesharing system for the present. Complete systems such as the one I have described will become available in the future; the technology is there and the benefits are there; however, it is impossible to determine the time frame.

In my opinion, becoming familiar with future business systems will allow the businessman to understand the forces at work and plan accordingly for the time in the future when businesses may need this type of system to survive and remain competitive. One of the best applications of the TRS-80 is to obtain experience with computers and at the same time use it to perform some of the minicomputer or microcomputer operations listed in my complete system in a cost effective manner.

(Editors note: John Strader is a Certified Public Accountant with an accounting degree from the University of Washington. For the past eight years he has worked for the State of Washington. He has audited over two hundred different accounting systems and performed audits on both large and small computer systems. He has recently opened a Certified Public Accounting office in the Olympia-Lacey Washington area.

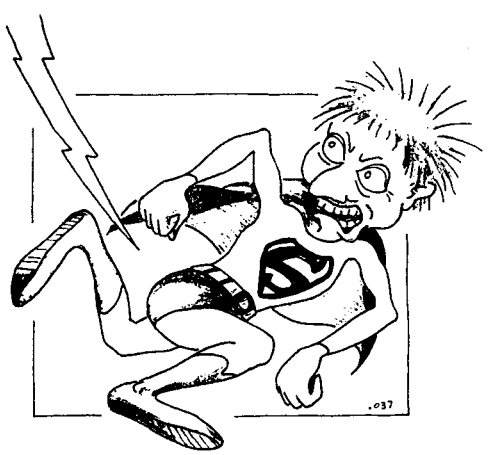
John has consented to conduct a business forum for the 80-US JOURNAL if enough reader interest is shown. Send us a postcard stating what type of business articles you would like to see in these pages: specific application articles, how to program and design specific systems, the type of hardware necessary, specific applications/use of the TRS-80 or any other subject you are interested in reading about.

We would also like to list the name and address (once) of any person or firm who sends in one business software program or users manual (space and time permitting). The program should be sent to the 80-US JOURNAL. 80-US will not publish, sell or market these programs without the permission of the programmer-owner. The intention of this service is to let our readers know of available business software which appears to be adequately prorammed and/or documented.)

HANGUPS

There have been several fixes for a problem which seems to plague everyone - keybounce. One is to get the Debounce tape from Radio Shack. We find that when using this one that we occasionally skip letters, sort of a "keybounce overkill". Another from Ruben Carril, Glendora, California is to bend back three of the four prongs, so that only one makes contact. We don't know if that will work, and in any case with this one you need to be very careful not to break them. Anyone tried this one yet? The one we have been using for about eight months now is to fill the entire inside of the key with Dow Corning Molykote (r) 4X lubricant. This makes the keys feel like velvet, reduces the "pingy" noise, and has reduced keybounce to virtually nil. The "gotcha" is that the lubricant gets into the bottom of the case and collects dust and generally makes a mess inside.

FOR THE LITTLE NYBBLERS



While out flying one stormy night
over the railroad yards,
Superman took a lightning bolt
right in the leotards.

With twenty-five hundred amps in the pants
and fifty megavolts there too, sad to say,
what resistance did those torrid tights offer
as our hero went steaking "Up, up and away"?

By L. B. Christopherson

80-U.S.
P.O. Box 7112
Tacoma, WA 98407

Bulk Rate
US Postage
PAID
Permit No. 774
Tacoma, WA